The Small Computer Magazine

kilobaud T.M.

Understandable for beginners . . . interesting for experts

August 1978 / Issue # 20 / $2.00 / DM 7,50 / Sfr 8,10 / Ffr 16,0 / UK £2

PUBLISHER'S REMARKS

Wayne Green

European Microcomputing

The May microcomputer show in Paris was well attended, even though it was a small show by U.S. standards. The *Kilobaud* booth was kept busy for three solid days handling subscriptions for eager Frenchmen.

Both Radio Shack and Commodore are well along in plans for manufacturing their systems in Europe, one in Belgium and the other in Germany. A small French firm, E.M.R., is making kit systems, and they are popular enough so I spotted a display right by the front door of the largest department store in Paris, Galeries Lafayette.

There are at least two manufacturers of systems in England so far and both seem to have their eyes set on the U.S. market in addition to the fast-growing market on the Continent. While in England I visited a newly opened computer store on the outskirts of London and also had lunch with John Marshall of NASCOM. The NASCOM exhibit at Paris was packed solid for the entire show.

There are presently about 20 computer stores in Europe, but the growth there seems to be paralleling that in the U.S., so watch out. The first issue of the London-based *Micro Computer World* magazine sold out its 25,000 first run, and their second issue seems to be going even better. Enthusiasm is high.

I was surprised and pleased to find *Kilobaud* well known in Europe . . . and a packed house for my scheduled talk at the Paris show. The room ws designed to hold an audience of about 500, but the aisles were also jammed with people, and they were crowded right up onto the stage. Many people had to be turned away.

Reader Responsibility

One of your responsibilities, as a reader of *Kilobaud*, is to aid and abet the increasing of circulation and advertising, both of which will bring you the same benefit: a larger and even better magazine. You can help by encouraging your friends to subscribe to *Kilobaud*. Remember that subscriptions are guaranteed—money back if not delighted, so no one can lose. You can also help by tearing out one of the cards just inside the back cover and circling the replies you'd like to see: catalogs, spec sheets, etc. Advertisers put a lot of trust in these reader requests for information. To make it even more worth your while to send in the card, a drawing will be held each month and the winner will get a lifetime subscription to *Kilobaud*!

The reader-service card of Michael Settle of Arlington TX was selected in our latest drawing. Mike wins a lifetime subscription.

More Software Needed!

Although the announced subject of my talk was the U.S. hobby-computing development, in actuality I discussed "The Fraud of Microcomputing." I pointed out that without the availability of programs for computers, their sale to consumers as devices that "can" do all sorts of wonderful things is effectively a fraud and a hoax. Even though these systems have the electronic capability to do all of these wonderful things, they are little more than expensive toys. In English we do not try to differentiate between the use of the word "can" as meaning capable of doing something in the future and capable at present.

This concept is not new to readers of *Kilobaud*, since I've been harping at this for about three years now. The recent article in *Money* magazine was justifiably critical of microcomputing. Many manufacturers and dealers got bent out of shape by the article, but all have to grudgingly admit that the criticism of microcomputers is valid. Until we have developed more programs that will actually permit microcomputers to do the things we say they can, they are going to have to remain strictly a hobby phenomenon.

EDITOR'S REMARKS

John Craig

Are You Ready for This?

I get excited about word-processing systems because they're near and dear to my line of work. And I get even more excited when a significant piece of software comes to the pages of *Kilobaud* and is made available for everyone . . . free! We have just such a piece of software for you this month. DOCUFORM! We all owe a vote of thanks to Donald Fitchhorn for developing it—and to Mits for deciding not to market it and allow its release. Just wait till you see some of the features this package has to offer. I'm sure you'll share my enthusiasm!

Like a Broken Record . . .

I really hate to sound like one (a broken record, that is), but I'm going to hop back up on my soapbox and jump on computer stores again. I keep hearing stories about how people go into stores to buy a system . . . and continue to be greeted with indifference, ineptness and a general lack of good salesmanship techniques.

kilobaud™

Kilobaud is published monthly by 1001001, Inc., Peterborough NH 03458. Subscription rates in the U.S. and Canada are $15 for one year and $36 for three years. In Europe: Kilobaud erscheint montatlich bei Fachzeitschriftenvertrieb Monika Nedela, 7778 Markdorf, Markstr. 3. Abonnement DM 70.- plus Porto 7.20 + gesetzl. MWST; Sfr. 81.- plus Porto 7.20. Australia: For subscriptions write—Katherine Thirkell, Sontron Instruments, 17 Arawatta St., Carnegie, Vic. 3163 Australia. UK: £20 sterling/year—LP Enterprises, 313 Kingston Rd., Ilford, Essex, IG1 1PJ, England. Please write for other foreign rates. Second-class postage paid at Peterborough NH 03458 and at additional mailing offices. Publication No. 346690. Phone: 603-924-3873. INCLUDE OLD ADDRESS AND ZIP CODE WITH ADDRESS CHANGE NOTIFICATION.

I said it in an editorial many moons ago, and I'll say it again: Just because a person is a sharp programmer or engineer *does not* mean that person can successfully sell computers! Granted, it might be easier for the person with the technical background to be trained in sales . . . rather than trying to go the other way.

The point is, there doesn't seem to be any of this training taking place! Why not? It's important. The majority of the people working in stores don't have any problem selling, or talking, to fellow computer enthusiasts. The problem arises when that businessman, or lay consumer, comes through the door.

The reason it's important that store personnel be *taught to sell* to these customers is because there are going to be more and more people coming into computer stores in the immediate future to inquire about, or buy, systems. *The general public is becoming aware of personal computers!*

You've seen those prime-time advertisements on nationwide television for the Tandy TRS-80, right? Have you also noticed how your friends and relatives have suddenly become aware of personal computing . . . and have begun thinking about it?

The side effects from such advertising will benefit the entire industry, and people are going to be looking at all the systems on the market. Naturally, they're going to be going into computer stores to accomplish this. They're going to buy . . . if they can be sold.

A mechanic would make as good a car salesman as anyone else . . . if he were trained in sales techniques. A computer programmer will make a good computer salesman . . . if he is trained in sales techniques. There are several approaches to this problem.

One is for the store owner to purchase a self-teaching salesmanship course that consists of cassette tapes and manuals . . . and then turn his people loose with it. (As a matter of fact, such a course could be used as the basis for developing a CAI course for the computer, and then be sold as one of the business packages available from the store!) Another method, and one which I think is very important, is to have the entire staff get together for role-playing . . . with one person acting as the customer and another as the salesman. Try it, and see if it helps!

Moving up . . . not out

Recently a couple of friends of mine have moved into newer and bigger facilities . . . and I thought I'd pass their addresses on to you. It always does my heart good to hear about such things because there are enough companies that *aren't* making it. Tarbell Electronics is moving to: 950 Dovlen Place, Suite B, Carson CA 90746. The old phone number is still good (213-538-4251), and they have an additional one also—213-538-2254. Mullen Computer Products has moved to 2306 American Ave., Hayward CA 94545. Their new number is 415-783-2866.

And speaking of companies . . . keep your ear to the ground and you'll be hearing a rumble approaching from the distance. That rumble is coming from Apple Computers! They have some exciting things coming up!

TROUBLE-SHOOTERS' CORNER

Ralph Wells

Some Fundamentals of TroubleShooting

Debugging microcomputers will be the subject of this column each month. Troubleshooting is often considered more of an art than a science. There are several fundamental guidelines, however, and we'll delineate a few each time. I'll try to use your problems as illustrative examples of specific techniques. Before getting specific, though, let's lay out a generalized, three-cornerstone foundation of the basics, namely: (1) assemble your "tools," (2) define the problem and (3) correct the problem.

Inventory Your Assets

The first step is to "take stock" . . . an inventory, so to speak, of what you have going for you. Your assets can be roughly divided into two categories: *knowledge* and *things*. It seems that there is never enough of either of them. The most valuable asset is knowledge—or perhaps its corollary, intelligence. Fortunately, you're probably well above average in these areas or you'd be reading *Hustler* instead of *Kilobaud*. The very fact that you're reading this sentence indicates that you're interested.

You're broadening your horizons now. You've probably been doing so for a long time, and the future is open for intriguing uncharted possibilities. If you haven't already discovered it, I'm sure you'll soon find that the field of personal computing attracts a unique type of mentality.

Naturally there are a lot of engineers and programmers involved, since these fields originally provided the foundation for the new era, but today they provide only an overrated nucleus. It is thinking people from other fields who fill out the ranks of personal computing. Get to know as many of them as possible, as soon as possible. You'll find that your common interest has eliminated dull, uninteresting people, leaving a cream of intelligence that comes in a fascinating array of human "packages." If there is a computer club around, try it!

Your Best Debugging Tool

One possible drawback of the "club" approach is that thinkers may not be gregarious joiners. This is unfortunate because no matter how little you may believe you know about your computer, I can guarantee that you can be of help to somebody else, who, in turn, can help you. What's more, you'll both enjoy it! *Your best debugging tool is knowledge*—yours *or* his!

A characteristic of the field of artificial intelligence seems to be that no one human has enough knowledge to cope with his own personal computer. A characteristic of the computer hobbyist seems to be that as soon as he attains even a marginal proficiency with his system (both hardware and software), he starts getting involved in something even more complex. He is again incompetent, presenting an even greater challenge to his learning processes. Now he needs help; he needs friends. I'm convinced that if the computerist were isolated, without *friends*, the whole movement would have died before it began.

When you buy an off-the-shelf computer today, there's a very good chance that it will work. Perhaps not as well as a similar purchase of a TV or hi-fi—but it *should* work. It wasn't always so. A couple of years ago, *none* of the first models off the production lines worked "all right." If you doubt this, check with any "old-timer." Incidentally, I don't know of any other situation in which an "old-timer" in a high-technology field may well be under 20 years of age. This is tied into another unique phenomenon of the time—the growth of the computer club.

Roots of Friendship

One day in the summer of '75, an envelope bearing the letterhead of a superior court judge arrived. I was relieved to find that my sins had not caught up with me: it was an invitation to join in the formation of a "computer club." The first meeting took place one hot Sunday afternoon in the "meeting hall" in Don Tarbell's apartment complex.

It featured an "imported" speaker (Bob Marsh) from the San Francisco Bay area who gave us an informal resume of the fledgling microcomputer clubs and related efforts "up North." It seems he also had a "garage-type" operation, and afterwards he set up a card table where I chatted with him and ordered a motherboard for my as-yet-undelivered Altair 8800. The hand-written receipt was rubber-stamped PROCESSOR TECHNOLOGY.

Later, several of us went over to the back bedroom of Don's apartment to see how he was developing a system to use ordinary (cheap) audio cassettes and recorders to store digital data.

For nearly a year I'd been developing an 8080 Dataworks system as part of my job at Inmarco, but that Sunday after-

noon was a turning point in my life. That was the day that computing became *personal* for me. It became *people* and not just *things.* It probably affected most of the others there the same way, because the Southern California Computer Society (SCCS), which was born that day, became an unparalleled phenomenon of proliferation.

Synergistic Synectics

The club members had much in common. Perhaps the most commonly shared problem was that no matter what you bought or who you bought it from, *it didn't work right.* Some things wouldn't work at all because of such difficulties as the write-enable bug and wiring diagram errors on the first Altair 8800s. It was usually only a partial failure, such as the double terminations on the first Processor Tech motherboards, the inverted diode in the Altair 680 and the failure of the KIM's 5 volt supply to reliably drive a Teletype.

It took a year and a half to get the intermittent devices out of my SWTP 6800, and I've given up completely on the Altair Computer II, but these are the unusual cases. I have a rather well-equipped garage and unlimited access to all of the facilities in the laboratory at Inmarco, but the most powerful tool of all has been *synergistic synectics.*

The English language occasionally supplies beautifully apt "old" words to describe new situations. Roughly interpreted, *synectics* means creative group problem solving, utilizing a variety of backgrounds and disciplines. *Synergism* means optimization through cooperation. Together, they define what I've been trying to describe—and it's powerful! Doctors, lawyers, housewives, stockbrokers, you name it—they are all part of a loosely knit problem-solving network the likes of which technology has never seen. I repeat, it's probably the most valuable asset in your problem-solving toolkit.

The Knowledge Explosion

Now let's make a distinction between *knowledge* and *intelligence.* As with most things, there's a trade-off between them, but intelligence has the advantage, since it will create knowledge. In fact, the intelligence of most hobbyists seems to create a veritable compulsion toward the acquisition of knowledge. Generally speaking, the areas you've had problems with in the past are the ones where you've built up the largest inventory of knowledge, and you're probably "good at it" now—or, at least, better.

Personal knowledge, accrued through either your own past experience or that of your friends, creates the fastest solutions to problems. There are other sources of knowledge. The magazine you're holding is probably the most effective. If you go into a computer store, you'll see something totally unique in the publishing industry: a complete (as possible) on-the-shelf display of back issues of every major magazine on the subject. Some even sell for more than the original price.

Why? The first copy of the SCCS club publication was little more than the kind of newsletter you'd receive from the local PTA. By the time my first article appeared (August 1976), it had 80 pages, a subscription list approaching 30,000 and a new name —*Interface Age. Byte* was off and running, I'd just subscribed to *Kilobyte* (now the magazine you're reading), and the venerable *Dr. Dobb's*, although still in a newsletter format, was becoming very thick.

Today the articles published monthly in magazines that didn't even exist three years ago outnumber all the articles published in all fields of electronics in *Electronics, Popular Electronics* and *Scientific American.*

Three years ago, the only books on the subject were the technical manuals published by semiconductor manufacturers. The hardware descriptions were written by and for advanced (very advanced) electronics engineers. As such, they were almost totally undecipherable by anyone else. The software tomes were worse. Even their own "hardware" men couldn't answer software questions, and vice versa. The Op-Amp Technical Bookstore in Hollywood opened last year to *specialize* in selling literature on this subject.

The sheer volume of literature now available is awe-inspiring. The books on programming in BASIC alone occupy a couple of shelves. I don't dare throw anything away because it's continually being used for reference to fundamental subjects, which probably explains the value of back issues.

In the realm of the printed word, we find ourselves inundated by an ocean of knowledge. If you spent 40 hours a week reading periodicals, I doubt that you could keep up. For this reason, I suggest that you limit your reading to those subjects of personal current interest, scanning the subheadings and subtitles of pictures in order to inform yourself of the *existence* of other information.

The Horse's Mouth

A third source of knowledge is the manufacturer of the product you're troubleshooting. This was originally a very effective technique, because it was a two-way street. The cause of your trouble was often a result of hasty design and inadequate debugging on his part, so the engineer-programmer-part-owner-janitor whom you spoke to on the phone (letters were seldom answered satisfactorily) was getting a lot of valuable feedback for free. I'm sure that if SPHERE had paid for the hundreds of customer man-hours spent in debugging their first units, at the wages their customers commanded on their "regular" jobs, they would have gone out of business a year earlier.

Most of the voices I came to know on the telephone belonged to very conscientious, hard-working, "turned-on" men with an insatiable enthusiasm for making their products everything they were claimed to be, and more. Today this still applies to most of the smaller suppliers, but attempts to get cooperation from some of the "grown-up" purveyors can range from frustrating to downright insulting.

Commodore, for example, would rather have you ship your PET back to San Francisco (the warranty says to allow two months for the trip) than to try to straighten out your problems by phone. True, there aren't as many bugs in the PET as in the early Imsais, Altairs, Jolts, etc., but my PET still needs a flea collar.

If you do call the factory, you'll get a lot more satisfaction for your phone bill if you've worked on the problem yourself and have it as well defined as possible. Gather all your schematics, printouts and test results in front of you before you dial. Remember that if you have a problem, it's very likely that others do too, and your contact may be a very busy man . . . make it short. Make a list of your questions beforehand and take a lot of notes, including the name of your contact. It's possible that top management may not share the indifference with which you are treated, nor be aware of the proficiency with which your problem is solved.

Another very effective method of getting unpublished knowledge from the "horse's mouth" is to go to the various trade shows and conventions that occur with increasing frequency. The displays are often staffed with very knowledgeable, key people, and the one-on-one atmosphere is most conducive to problem solving. Both sides benefit.

So far I've spent a lot of effort delineating the roles of intelligence, synergistic synectics and knowledge as they relate to debugging artificial intelligence. An entire book would be required to do justice to the subject—I can't overemphasize its importance. This may be disappointing to those who expected a dissertation on scopes, DVMs, pulse probes, etc. These things are important, but they are just that—*things.* In many fields, things are more valuable than people, but not here, not today.

The Test Bench

Judging from my observations, there seems to be a direct trade-off between knowledge and test equipment. To a lesser extent, this is often true when the problem is one of software. If a software bug is dynamic, then "state analysis," "signature analysis" or "time domain" techniques using storage scopes and/or equipment such as the Paratronics logic analyzer or the Phoenix digital signature analyzer may be the only practical solutions.

Several years ago I was amazed to watch a rather complex piece of audio electronics being serviced with only a clip lead and an electrolytic capacitor. It was done on a well-equipped test bench in the sound department of Columbia Pictures, where I was working at the time. All of the sophisticated (expensive) test gear we normally used was available, but by using only 50¢ worth of "test equipment," the troubleshooter had the problem isolated and the defective part replaced in less time than it would have taken to warm up the $3000 oscilloscope I would have used.

I'm certain I'd have found the bug eventually, but I would have used half the equipment on the

(continued on page 19)

BOOKS

Programming a Microcomputer: 6502
Caxton C. Foster
Addison-Wesley Publishing Co.
Reading MA, $8.95

Programming a Microcomputer is one of the few really good books I have seen on any microprocessor, let alone the 6502. It is a well-written exposition on programming the 6502 microprocessor and covers just about every instruction except the operation of the BRK (break) instruction. I believe MOS Technology (Commodore) should include a copy with every KIM-1; it is the perfect companion to their 6502 manuals and any good basic hardware book such as Osborne's *Introduction to Microcomputers,* Vol. I.

The book assumes a KIM-1 is at hand, but the material can be easily (or maybe not so easily if you don't have a hexadecimal keyboard/display or a Teletype) adapted to any 6502-based microcomputer with a 6530 I/O and interval timer chip. The chapter on the console controls is the only chapter that deals specifically with the KIM-1. Push-button switches can be substituted for the KIM-1's keyboard in the one project that makes specific use of the keyboard.

The flow of the book is not immediately obvious. At first glance, it seems to be a "Ten-Projects-For-Your-KIM" type of presentation. Reading it in detail, I was surprised to find that the projects are primarily vehicles for discussing the 6502 instructions and programming techniques. Flowcharts and diagrams are used effectively in explaining the operation of the instructions and describing the programming steps in each project.

Chapter 1 explores how computers operate by using the analogy of a clerk shuffling papers between IN and OUT baskets on his desk. For a newcomer to the field, linking computer concepts to a familiar model makes comprehension much easier. This chapter is an excellent jumping board to a book like Osborne's which goes into much more detail. Appropriately enough, the clerk analogy is referred to less and less as the concepts introduced become clearer.

The book becomes really interesting beginning with Chapter 3, which describes a Morse code oscillator using the KIM-1, a switch, three resistors and a small speaker. Foster introduces the concept of looping, describes relative branching and the 6502 branch instructions and uses this foundation to show the reader how to generate tunes using a loop.

Music buffs will find the piano-keyboard and tune-player chapters fascinating. Table lookup techniques are explained, and Foster even includes the tunes from "Red River Valley," "British Grenadier" and "Nearer My God To Thee" as incentives. For train engineers, on the other hand, Foster offers MICROBART (BART—San Francisco Bay Area Rapid Transit), which controls two trains approaching an intersection so that collisions can be avoided.

The last two chapters discuss the construction of a simple interpreter and assembler for a "dream machine" using an invented set of 15 commands. It's too bad these commands aren't 6502 instructions, but the parallels are obvious and the flowcharts are a good starting point for writing your own if you enjoy that much work.

The book has been interesting up to this point, but not so much as to deserve the praise I lavished upon it in the beginning. The clincher is that cached in the appendix are the programs for *every project in the book,* including the interpreter and assembler. This considerate addition to the book is unusual because the typical approach is to leave the specifics up to the reader, and this is *not* the approach to take toward beginners.

The only books I ever consider recommending are those I find indispensable, and this is one for all 6502 owners. Lest some of you get the impression that I am a beginner also, I have been programming for nine years now in machine language, BASIC+, COBOL and FORTRAN IV, so unless a book offers fresh and interesting insights, I usually regret paying for it.

Clint O'Connor
Crawfordsville IN

The Universal Elixir and Other Computing Projects Which Failed
Robert L. Glass
Computing Trends
Seattle WA
79 pages, $7.50

The Universal Elixir is a mildly entertaining book about some of the computer world's dismal failures. Projects fail for a variety of reasons, and this book was written to illustrate those reasons.

The author tells us that all of the tales are true, although names and places have been changed to protect those involved. I'm sure that anyone who works with computers in the business world or in government will identify with these situations. There are stories here about managers who are ignorant of computers in general, hardware experts who ignore software considerations and software wizards who blame failures on poorly designed hardware.

The vast majority of computer books are technically oriented. I suppose it was inevitable that the industry would some day begin to generate its own nontechnical literature. The only problem is that the market for this sort of book will be quite small, compared to the millions who might purchase the latest Harold Robbins effort. And so, we find a $7.50 price tag on a 79-page book. That's a lot to shell out for half an hour's worth of reading. Perhaps we should all wait for *The Universal Elixir* to appear on the shelves of our local libraries.

Jeff DeTray
***Kilobaud* staff**

The Z-80 Microcomputer Handbook
William Barden, Jr.
Howard W. Sams & Co., Inc.
Indianapolis IN
304 pages, $8.95

If you're like me, you may have thought that Zilog's Z-80 microprocessor was more or less a simple expansion of the 8080 theme. Well, that's true to a certain extent, but the Z-80 is much more than just a super 8080. Just how much more is revealed in *The Z-80 Microcomputer Handbook.*

The book is divided into three major parts. Section I is a complete description of the Z-80 hardware, including such hardware-dependent topics as the instruction set and modes of addressing. You begin to get an idea of the true potential of this microprocessor when you discover there are 158 instructions (as opposed to 78 for the 8080 and 80 for the 8085), 14 general-purpose registers and 10 distinct modes of addressing. Section I will also tell you about timing, flags and interrupts. It concludes with a discussion on how to interface memory and I/O devices to the Z-80.

Software is the business of Section II. Now you get a look at just what the expanded instruction set and all those registers can mean to you in terms of programming. The Z-80 has much to offer in the areas of logical and arithmetic shifts to CPU registers and memory, the handling of character strings and data transfer and I/O. Some commonly used subroutines are presented at the end of Section II. This thoughtful addition saves you the time and effort required to duplicate these often needed utility routines.

Section III tells you how to go about the task of building a complete system around the Z-80. First, the many bits and pieces offered by Zilog are enumerated and described. Zilog offers a Z-80 development system consisting of CPU, dual floppy disk drives, 16K RAM, and a software package that includes a debugging package, operating system, editor, assembler, and file maintenance. Of course, Zilog isn't the only source of Z-80 hardware and software. In fact, the Zilog system is targeted primarily for commercial users. For that reason, the author also includes information about TDL, Cromemco, The Digital Group and even Radio Shack systems using the Z-80 microprocessor.

Perhaps you've been trying to determine whether or not the Z-80 is the processor for you. The only way you'll ever make an intelligent decision on that matter is to really learn something about the Z-80 and its capabilities. *The Z-80 Microcomputer Handbook* will enable you to do just that.

Jeff DeTray
***Kilobaud* staff**

Read a good one lately? Why not inquire about reviewing it?

LEGAL/ BUSINESS FORUM

Kenneth S. Widelitz
Attorney-at-Law

Procedures for Starting a Small Business

Starting your own small business is really not that difficult. All it takes is a good idea, some initiative, a sprinkling of common sense and a little working capital. You would be surprised in some cases how little money it actually takes.

I recently had occasion to start my own small business and will use that situation as a framework for this discussion. In September 1977, I had my system up and running and was in the market for a quality word-processing printer. I didn't want to fork out approximately $3300, which seemed to be the going rate for a good, new daisy-wheel printer.

I turned my attention to the used computer equipment market. I quickly found that while such a market existed, there was a paucity of high-quality hardcopy devices in the marketplace. I did find some reconditioned Diablo mechanisms, but what I had in mind was a full-fledged printing terminal. I didn't want to have to build my own cabinet, power supply, interface, etc.

After looking for three or four months, I finally found a leasing company that had reconditioned Qume- and Diablo-based terminals. This was exactly what I wanted and I put in an order for one. Just out of curiosity I asked how many were available. When I was told there would be about 200 terminals coming off lease during the next year, something clicked in my mind. The idea for my business was born. Surely, there were other hobbyists and small businessmen who needed a quality word-processing printer but who couldn't afford, or didn't want, to buy a new one.

That is the way COMPUTER TEXTile got started. What follows is a description of how to turn your idea into an ongoing small business.

The Preliminaries

Once you have your idea, if you are anything like me, you want to have it implemented yesterday. Well, that's just not possible. But in order to get your business going in as short a time as possible, it is important to map out what needs to be done and approximately how long it will take to accomplish each task. Common sense dictates that the things that will take the longest should be started the soonest.

You need a location for the business, a phone, stationery, literature describing your goods or services, a slew of licenses and permits, insurance, a bank account, press releases, perhaps the capability of making sales using Master Charge or VISA, among a myriad of other considerations.

Obviously, the first thing that needs to be done is to find a location for the business. Depending on the business, a spare bedroom, garage, storefront, small office or warehouse may be appropriate. In my own situation a place of business was not a problem as I already had an office for my law practice.

After establishing a location, you should immediately order a telephone. Ma Bell is one of my pet peeves for many reasons, not least of which is the length of time it takes for her to give you a phone line. Apart from that, you need your phone number before you can go to a printer to have your stationery prepared. You also need the address and phone number for your bank checks.

Among the preliminaries is establishing and documenting your legal relationship with the people with whom you will be dealing. If you are going to have a partner or partners, you should have a partnership agreement prepared and signed before you spend any money.

Since COMPUTER TEXTile was a sole proprietorship, I only had to worry about entering into written agreements with my supplier of the terminals. We executed a Right of First Refusal by the terms of which I acquired the right to purchase every terminal that came off lease. I also entered into an agreement that required the leasing company to recondition and be responsible for the working condition of the units. Obviously, the types of agreements you will have to enter into depend on the type of business you are starting.

Licenses and Permits

The requirements of doing business in any particular city, county or state vary greatly from place to place. I will discuss the specifics I ran into in Los Angeles, as I feel its requirements are typical of those found throughout the United States.

If you are doing business in a name other than your own, it is necessary to file a Fictitious Business Name Statement, or whatever your particular state happens to call it. In other words, it was not necessary for me to file such a statement when I opened my law practice, as I was doing business as Kenneth S. Widelitz. On the other hand, in this new endeavor, I am doing business under the name of COMPUTER TEXTile. Legally the business is referred to as Kenneth S. Widelitz dba (doing business as) COMPUTER TEXTile.

In California the fee for filing the Fictitious Business Name Statement is $10. That's not really what it costs. The statute requires that the Fictitious Business Name Statement be published in a newspaper once a week for four successive weeks. The statement should be published in a newspaper of general circulation in the county where the principal place of business is located. Generally such statements are published in the local legal newspaper.

I had mine published in the *Daily Journal*. Their fee for publication and filing the statement with the county clerk was $25. The filing of the statement with the county clerk raises a presumption that you are doing business in the format indicated on the statement (sole proprietorship, partnership or corporation) and prevents another business from using the same name in the same county.

Next you will probably have to acquire a license to do business in your municipality. In Los Angeles what is technically required is that you register to pay a business tax. Personally, I can think of many things I would rather do than to stand in line waiting to register to pay a business tax.

The Los Angeles business tax breaks down into three categories: Retail Business Tax, Wholesale Business Tax and Professions and Occupations Tax. The first two types of business tax are self-explanatory. The third covers people, such as system consultants or attorneys, selling services. It is possible that a single business will require all three business tax licenses.

COMPUTER TEXTile required the Retail and Wholesale Business Tax permits. In Los Angeles, the tax is based on gross sales. That strikes me as being grossly unfair since businesses with small gross margins but large dollar-volume sales are penalized.

In Los Angeles the Retail Business Tax is $18.75 for gross sales up to $15,000 and $1.25 for each additional $1000 of gross sales. The Wholesale Business Tax is $20 for gross sales up to $20,000 and $1 for each additional $1000 of gross sales. The Professions and Occupations Tax is considerably higher. The tariff is $30 for the first $6000 in gross fees and $5 for each additional $1000 in gross receipts. The fee for the base amount is all that need be paid up front for the issuance of the Tax Registration certificates.

If you are dealing in reconditioned or used equipment, you will probably have to get a Police Permit, which, in Los Angeles, costs $105. The cost is not the major objection to that permit. The killer is that all used equipment must be held for 21 days at the place of business. When I read that, my immediate thought was that it might very well prevent me from going into business. I just didn't have the capital to inventory a number of $2000 items for 21 days. It was obvious that I needed to find a loophole.

A discussion with a police investigator showed me the way. I explained that in the ordinary course of my business I never actually see the terminals. Rather, I instruct the leasing company to send the unit directly to my customer COD. How could I hold something for 21 days that I never actually had? The investigator said that in such a case I was actually a procurer rather than a retailer or wholesaler. Who was I to argue with that? I decided that it was unnecessary for me to acquire a Police Permit.

For those who are wondering why such a permit exists, it is actually aimed at helping the police to recover stolen merchandise. One of the rules is that all used merchandise acquired must be reported to the police department. During the 21-day period they can check to see whether it conforms to a description of stolen goods.

A very important permit is the Resale Permit. In California it is issued by the State Board of Equalization. The Resale Permit allows you to buy merchandise to be resold without having to pay a sales tax. The holder of the permit is responsible for collecting the sales tax from the end user.

While there is no fee for the issuance of the Resale Permit, the State Board of Equalization does require security for the sales tax you are to collect for them. That is, they require that the holder of the permit deposit with them either a certificate of deposit, a cash deposit or a surety bond in a specified amount equal to the estimated sales tax to be collected for six months.

In my case, the security was well over $1000. Not wanting to tie up capital in a security deposit, I opted to deposit a surety bond. In general, the cost of a surety bond (premium) is 10 percent of the face amount of the bond. To minimize the security costs it is a good idea to be rather liberal in your estimate of out-of-state sales and wholesale sales, both of which are not subject to sales tax, and to be rather conservative in your estimate of retail sales on which a sales tax must be collected.

The Business Tax licenses and Resale Permit can be acquired in a day, at least in Los Angeles. From a commonsense point of view you should take care of these items after making arrangements for those considerations that take longer.

Master Charge/VISA

If you are going to be doing any retailing, you will probably want the capability of allowing your customers to use Master Charge or VISA cards. It takes about two weeks for your bank to set you up. The bank's fee, its discount rate, is a function of average ticket size and average monthly volume. The discount rate varies from 3 percent down to 1.25 percent. To qualify for the lowest rate your average ticket will have to be over $50 and your average monthly volume will have to be well over $5000. Of course, these rates vary from bank to bank and you would be wise to do some rate shopping.

Other Considerations

One of the first things you would want to do is to let people know that you are in business. Depending on the type of business, the way in which you will do this will vary. In my own situation I immediately issued new-product releases announcing that there was now a source of reconditioned high-quality hardware available to the hobby and small-business market. Remember that the lag time between your new-product announcement and its appearance in a magazine can be three or four months, so this is one item to take care of as soon as possible. Don't do anything dumb like announce a product before it is actually in existence.

If you have an employee it is necessary to obtain both a federal and state employee-identification number. Many states also require that you have Workman's Compensation Insurance once you have one or more employees. You should talk to your insurance agent about other insurance needs such as liability and products liability insurance.

Depending on your location and type of business, there may be other requirements that I have not mentioned. You can obtain information regarding such requirements from the business tax division for your state's equivalent of the IRS and from your local municipality's license bureau.

Good luck in your entrepreneurial endeavors. (By the way, anyone interested in a quality word-processing terminal at a great price?)

Warrantor of the Month

Philip K. Hooper of Johnson VT has nominated A B Computers of Perkasie PA as warrantor of the month. Mr. Hooper wrote that he purchased a KIM from A B last summer and experienced some minor trouble with the keypad. A B offered and actually provided a "loaner" KIM while his own was being serviced. Mr. Hooper's KIM was returned fully repaired within two weeks. Any other nominations for Warrantor of the Month?

KB CLUB CALENDAR

Steve Fuller

Roosevelt NJ

There's a move afoot here to start an EMPL users group/newsletter. If you're interested in contributing an article or a program, contact Erik Mueller, PO Box 17, Roosevelt NJ 08555.

Plainview NY

The STACK is the monthly publication of the Long Island Computer Association (LICA). Dues are $10 per year, and information is available from LICA, 36 Irene Lane E., Plainview NY 11803.

Omaha NE

For one SASE per issue, you can receive the newsletter of interest to SC/MP and SC/MP II owners. Primary objectives of the newsletter's organizer include formation of a software/hardware library and a bibliography of information sources that will be available to members at reproduction cost.

Write to Tom Bohon, 2215-A Walker Drive, Omaha NE 68123.

Boulder CO

Hillel Segal has been elected president of the new Association of Small Computer Users (ASCU). The group plans to provide members with selected publications at reduced cost, a bimonthly newsletter and information exchange and benchmark comparisons of competing small computer systems, according to Segal.

Membership fees will be $25 per year for current or prospective users of small computers, and will include a number of reports and periodicals.

Write to the Association of Small Computer Users, 75 Manhattan Drive, Boulder CO 80303.

Mobile AL

Hobbyists in Mobile have recently formed the Greater Gulf Coast Computer Club (G^2C^3). Meetings are held on the first Wednesday of every odd month and feature discussions on a wide variety of computer-related topics. Details are available from Marty Sharik, Greater Gulf Coast Computer Club, 512 Tuttle Ave., Mobile AL 36604.

Los Angeles CA

Got an Elf in your closet? Write to Patrick Kelly, PO Box 7162, Los Angeles CA 90022, for information on the COSMAC-1802 Users Group.

San Antonio TX

The Alamo Computer Enthusiasts meet here on the fourth Friday of each month in Room 104 of the Chapman Graduate Center, Trinity University. For details, write to the club at 7517 Jonquill, San Antonio TX 78233, or call (512) 657-3069.

Canoga Park CA

Jim Zuber of 20024 Cohasset #16, Canoga Park CA 91306, would like to hear from anyone interested in forming a KIM-1 users group in the San Fernando Valley area. You can call him at (213) 341-1601.

Philipsburg PA

Transaction is a newsletter for PET owners. If you'd like to contribute ideas or articles for this bimonthly publication, send them to PO Box 461, Philipsburg PA 16866.

(continued on page 20)

NEW PRODUCTS

KIM Expansion—the Atwood Way

There are several memory-expansion techniques available for the KIM today. One of the most versatile is the system being offered by Atwood Enterprises. Let me elaborate on that versatility.

The interface board, shown in the photo plugged into the KIM expansion connector, not only provides the decoding for additional 4K memory segments, but also offers write protect for four of those 4K blocks. Because of the ribbon cable used to connect the motherboard to the interface board, a degree of versatility is obtained with regard to component placement. This can be an important consideration if the KIM is mounted in a cabinet, and it is desirable to have the motherboard situated away from the KIM.

Going beyond memory expansion, versatility also is obtained from having a variety of I/O, analog, EPROM and PROM boards to choose from (manufactured by Atwood).

The 8-slot motherboard (shown in the upper-left corner of the photo) has a unique layout that permits the 4K memory boards to be inserted in either direction. The motherboard is double-sided with large areas of ground plane opposite large areas of voltage plane on the opposite side. This has the effect of making most of the board a capacitor to help suppress noise on the 5 volt lines.

Atwood Enterprises has been offering their 4K memory board (lower left corner of photo) for over two years. From the beginning, as well as now, the board is a good buy for the price and goods received: $89.95 assembled and tested. The board doesn't come with sockets, so you can purchase it in kit form ($79.95) and socket everything.

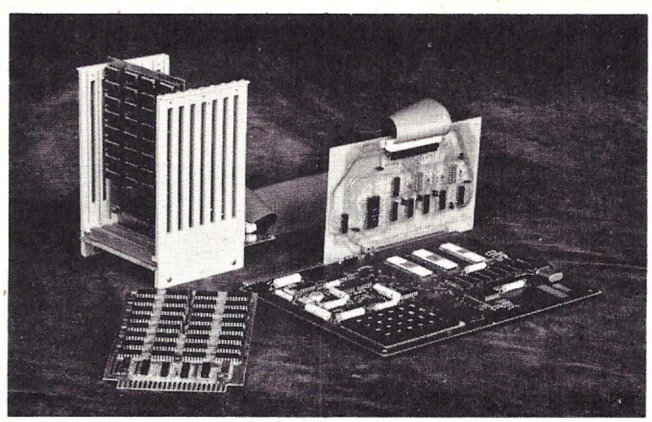
The interface board sells for $24.95 and provides 4 board select lines for four 4K RAM boards on the motherboard. Nine additional lines are available on the 50-pin connector for further decoding of additional RAM boards. As mentioned before, there are also lines, which can be brought out to switches, for write protecting any one, or all, of the 4K segments.

One of the definite advantages of going with the Atwood system is that there is a variety of boards available for expanding a KIM to a full-blown system. In addition to the 4K RAM board, Atwood also has a serial I/O board with seven software-controlled serial ports that can run up to 500 baud (price, $129.95). They have a 2716 PROM programming board that sells for $49.95 and a 2716 8K board that sells for $149.95. Getting back to the I/O department, they also offer a Digital I/O board with eight parallel ports, plus 16 interrupts, for $59.95. Their analog input board handles 32 inputs, eight bits each, and sells for $99.95.

Model 1500 Message Center.

For further information, contact: Kathryn Atwood Enterprises, PO Box 5203, Orange CA 92667.

Bill Lewis
1993 Elmwood Dr.
Santa Maria CA 93454



The Atwood KIM memory-expansion system.

New Ford Code-A-Phone Model 1500 Message Center

GRS Instruments, Inc., is now marketing the new Ford Code-A-Phone Model 1500 Message Center: the first telephone answerer intended for home or office use that comes with its own styled telephone built into the unit. It's easy to operate and install, and it is not subject to telephone company tariffs or monthly service charges. It is FCC approved.

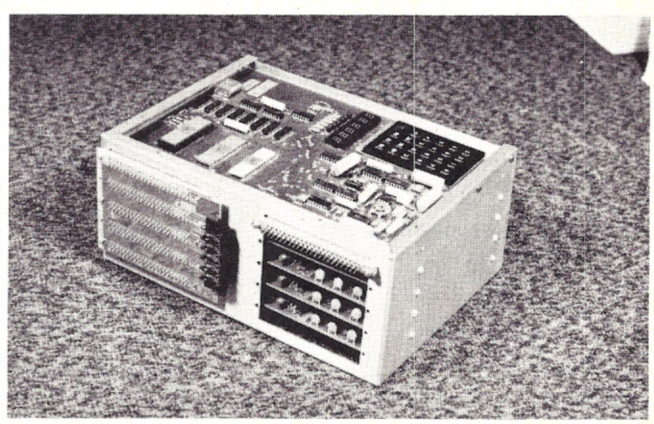
The Message Center permits outgoing announcements of up to 20 seconds and records up to 20 incoming calls of 30 seconds each. It features an exclusive lighted Call Counter so you'll know how many calls you have received and which call you are listening to. No-touch Call Screening lets you hear your caller's voice *before* you answer; Fast Forward and Rewind will let you find any message in seconds. It will also let you selectively erase any one or all of your messages at any time. Its Remote Control feature will let you hear your messages from any telephone in the world.

The Message Center is available in two color combinations; black with leather, and beige with walnut. Each sells for $299.95 plus $2.50 shipping and handling.

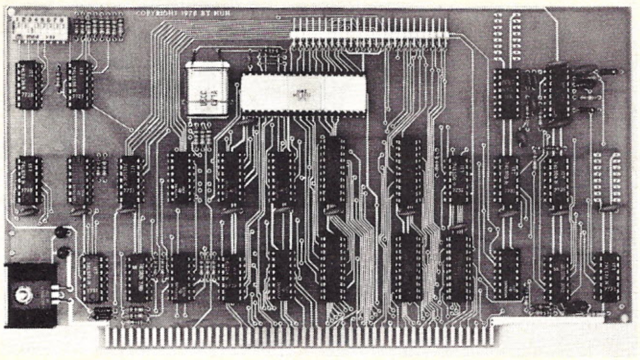
GRS Instruments, Inc., 8730 King George Dr., Suite 100, Dallas TX 75235.

Card File for KIM-1 Microcomputer

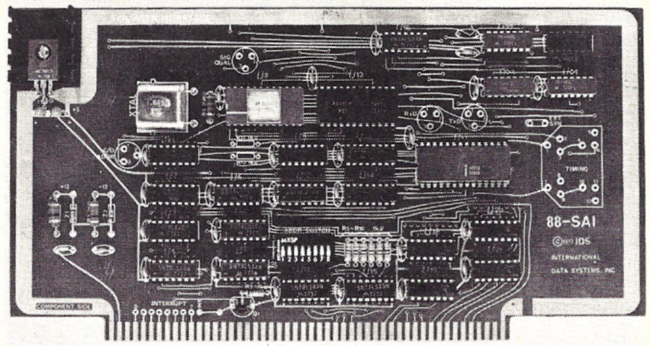
Micro Technology Unlimited, PO Box 4596, 29 Mead St., Manchester NH 03108, has introduced a compact card file with integral backplane board for KIM-1 system expansion. Even though overall measurements are only 5 by 9 by 11.5 inches, the unit holds a KIM-1 and up to 4 KIM compatible expansion boards. The key to its compactness is that expansion boards are mounted parallel to and underneath the KIM rather than perpendicular and off to the side as



Micro Technology's KIM-1 card file.



HUH's S-100 MPA.



88-SAI.

they are in similar products. Besides using no more table space than the bare KIM, the expansion boards are hidden away from prying fingers and other hazards.

Because the standard KIM bus is unbuffered, special attention was given to layout and shielding of the backplane board. With an electrical length of only 4 inches, nearly all of the bus drive capability can be used by the expansion boards rather than being wasted driving long, high-capacitance bus lines. Thus most boards designed for direct connection to the KIM's expansion bus will work successfully with the K-1005 card file. Power input to the KIM and expansion boards is conveniently prewired to a 5-point terminal strip.

Construction is of rugged 1/16 inch corrosion-protected aluminum. The backplane uses industrial-grade 44-pin edge connectors soldered to a two-sided fiberglass printed circuit board. The price is $68 completely assembled with six prewired connectors.

S-100 Adapter for Commodore PET

HUH Electronics' S-100 MPA, an S-100 sized card, plugs into the user's mainframe, and a cable connects to the PET, allowing the use of the wide range of peripheral and memory cards available for the S-100 bus. The S-100 MPA (Memory and Peripherals Adapter) is unique in that it emulates the true S-100 bus, including full DMA, true PSYNC generation, I/O address mirroring, read wait states and much more. MPA meets most of the proposed IEEE Standard for the S-100 bus.

An important feature of this versatile board is that it can also act as a stand-alone 6502 CPU board for the S-100 bus. It is the only 6502-based processor board to be truly S-100 bus compatible. A simple option kit is all that is required.

The S-100 MPA is available in kit form for $199.95, or fully assembled and tested for $279.95. The stand-alone processor option is an additional $49.95.

HUH Electronics, 1429 Maple St., San Mateo CA 94402.

Full Operating System in a Single Unit

Computer Data Systems of Wilmington DE announces their disk-based computer systems, the VERSATILE 3B and their expanded version, the VERSATILE 4.

This single unit computer combines a 9 inch video screen with 24 x 80 display, built-in minifloppy disk drive with 143K bytes of storage, upper/lowercase alphanumeric keyboard, separate numeric keypad and all electronics within a durable plastic enclosure.

The computer mainframe incorporates the 8085 CPU, 24K static RAM and a serial I/O port with RS-232 connector. The VERSATILE 4 expands on this system by providing 32K static RAM and 315K bytes of storage.

Operating software is supplied with both units and includes 20K Extended BASIC by Micropolis, a disk operating system and a complete software library of demonstration programs. Five diskettes comprise the Software Library, which includes several games, and a small-business accounting package. Two disks are blank so systems users may enter their own programs.

Access to Computer Data System's Software Library is open to all purchasers of either of these computer systems. Library listings are readily available at no charge, and programs may be requested on diskette for a copying charge only.

Computer Data Systems, 5460 Fairmont Drive, Wilmington DE 19808.

SAI Interface

International Data Systems, Inc., announces the 88-SAI (Synchronous/Asynchronous Interface) for S-100 bus computers. The 88-SAI, which provides a synchronous or asynchronous port for any S-100 bus processor, is intended for use in special communications requirements such as synchronous communications between S-100 computers and large-scale computers, high-speed modems, data encryption devices or other S-100 computers.

The 88-SAI allows baud rate, word size, parity and number of stop bits to be selected completely under software control. Also under software control is synchronous/asynchronous mode selection and functions associated with synchronous communications such as number of sync characters.

Fully compatible with RS-232C interfaces, the 88-SAI has additional provisions for interface to nonstandard devices requiring that various signal or handshake lines be inverted. The 88-SAI also provides interface to MIL-STD-188 level devices.

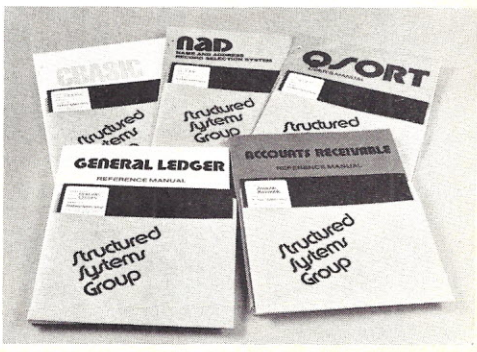
The 88-SAI is manufactured from G10 epoxy glass, measures 5 x 10 inches, not including edge connector. It requires four consecutive I/O addresses, and the board address is selected by a DIP switch . . . available in kit form for $199, or assembled, tested and with a limited warranty for $299.

International Data Systems, Inc., 400 North Washington St., Suite 200, Falls Church VA 22046.

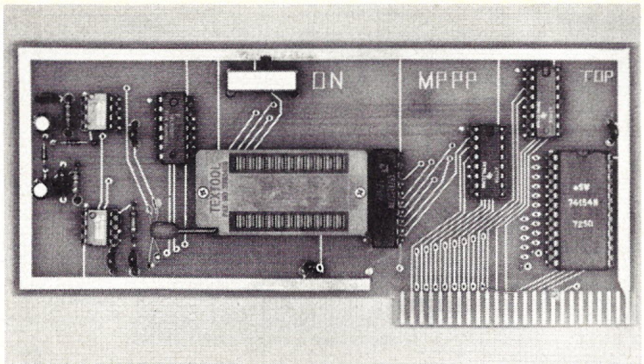
SSG Accounts Receivable System

Accounts receivable can now be processed on your 8080- or Z-80-based CP/M microcomputer system. Structured Systems Group has just announced availability of its general ledger compatible accounts receivable system.

Computer Data Systems' VERSATILE unit.



SSG's business software.



Apple II EPROM Printer.

The software package, latest addition to SSG's growing Business Systems series, is a full open-item billing system for small businesses. The programs are designed for use by business-oriented professionals who need reliable and simple operations. The system is excellently documented with a 120-page reference manual that quickly orients the first-time user and provides depth for the experienced hand.

The A/R system, priced at $750, features itemized statements, two aged trial balance reports, late charges, reminder letters, recurring receivables, sales reports and more.

Structured Systems Group, Inc., 5615 Kales Ave., Oakland CA 94618.

Apple II EPROM Programmer

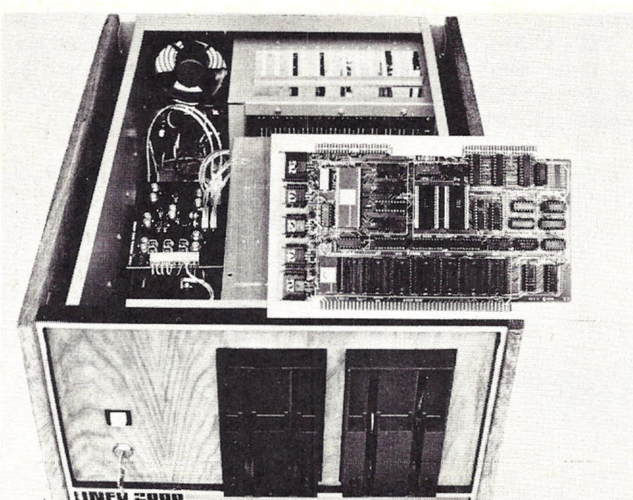
Microproducts' new Apple II EPROM Programmer takes the lid off the capabilities of microprocessor permanent memory. The two empty ROM sockets can be filled with 4K bytes of user selected programs. No more loading of peripheral interfaces, assemblers or your favorite programs or games from tapes. Merely turn on your computer and go!

For more advanced programmers, the permanent software options are almost limitless. The entire 8K of Apple II monitor and BASIC can be replaced with any operating system that can be put in Apple II compatible ROM. Systems similar to FORTRAN IV, FOCAL, ALGOL, COBOL, etc., can be developed and plugged directly into the 12K byte sockets presently wired into the Apple II. With the addition of "plug-in" ROM boards, 84K bytes of additional permanent memory can be added to your Apple II.

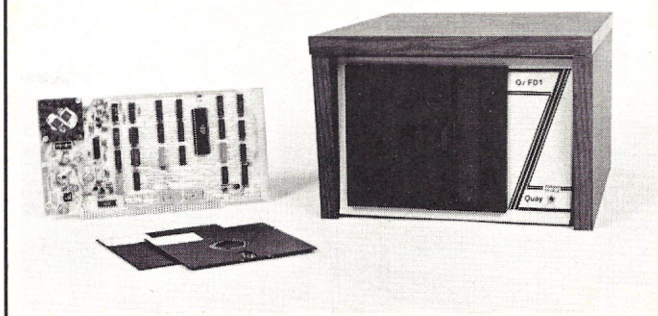
The Microproducts/Apple II EPROM Programmer consists of a fully assembled, double-sided, fiberglass printed circuit board with plated-through holes and gold-plated edge connector. It plugs directly into any available slot on your Apple II board. The programmer is fully self-contained and has its own "on-board" 25 volt power supply for programming the INTEL 2716 EPROM.

After your EPROM is programmed, it can be plugged directly into your Apple II empty ROM sockets with the Microproducts/INTEL 2716 socket adapter. Price of the programmer is $89.95; the INTEL 2716 socket adapter is $9.95.

The Microproducts/Apple II EPROM Programmer can be purchased from your nearest Apple dealer, or contact Microproducts, 1024 17th St., Dept. K, Hermosa Beach CA 90254.



INFO 2000 disk system.



The Quay 80 F1.

Floppy-Disk System for All S-100 Bus Microcomputers

INFO 2000 Corporation's new high-performance floppy-disk system incorporates a controller board called DISCOMEM, which was originally developed for use in the INFO 2000 Business System. This controller enables the manufacturer to offer much faster disk performance while lowering the overall cost of the disk system by $400. The S-100 disk system combines the PerSci Model 277 dual diskette drives with the INFO 2000 DISCOMEM Controller Board and Digital Research CP/M to provide all necessary hardware and software, when added to any S-100 bus computer, for immediate operation. Two spindles accept standard soft-sectored 8 inch flexible diskettes. The system provides full compatibility with IBM 3740 format.

The PerSci drives used in the INFO 2000 Disk System have voice-coil positioning. This provides seek times up to eight times faster than other drives using stepping motors for positioning. The DISCOMEM Controller is especially designed to take maximum advantage of the high performance capabilities of the PerSci drive.

In addition to the disk controller, the DISCOMEM board contains input/output interfaces required for most microcomputer systems. These include 2 RS-232 serial interfaces with software selected baud rates from 50 to 19,200 bits per second, 3 8-bit TTL-level parallel interfaces (2 output, 1 input), and provision for 8K of EPROM or for 7K of EPROM and 1K of scratchpad RAM. The I/O facilities enable the DISCOMEM to be combined with just two additional S-100 logic boards—a CPU board and a 32K RAM board—to create a complete high-performance, disk-based microcomputer system for business or scientific applications.

Cost of the complete dual-drive disk system, including all I/O facilities and CP/M is $2600. Another model, without the I/O facilities, is available for $2450.

INFO 2000 Corporation, 20630 S. Leapwood Ave., Carson CA 90746.

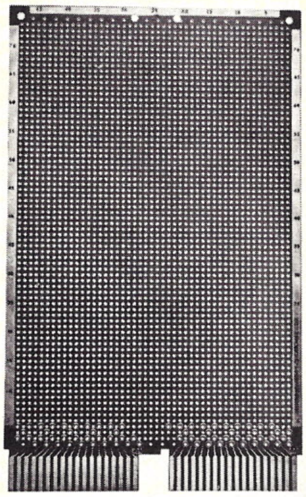
S-100 Floppy-Disk System

A floppy-disk system for use in S-100 bus computers is available from Quay Corp., PO Box 386, Freehold NJ 07728. The Quay 80 Fl system, priced at $795, includes the Q/80 FDC—floppy-disk controller board (capable of supporting up to four disks), QDOS—disk-based operating system, the Q/FD1 125 KB 5¼ inch band-driven disk drive with power regulator and interface cable, and the Q/80 FC—floppy-disk cabinet. Add-on drives (Q/FD1) are priced at $395 each.

In addition to the floppy-disk support, the Q/FDC has available a programmable 8-bit, TTL compatible, parallel I/O port capable of supporting standard peripheral devices such as line printers, tape punches, keyboards, etc.

NORTHSHARE

A time-share disk BASIC sys-



Vector's prototyping card 4607.

tem is now available for users of the popular North Star Floppy-Disk System. Designed to operate with either 8080 or Z-80 processors, NORTHSHARE provides up to four independent users with selectable memory partitions and buffered terminal outputs.

Minimum memory requirements for operation are 24K bytes. There are no special hardware requirements outside of additional terminals and I/O ports to support multiple users.

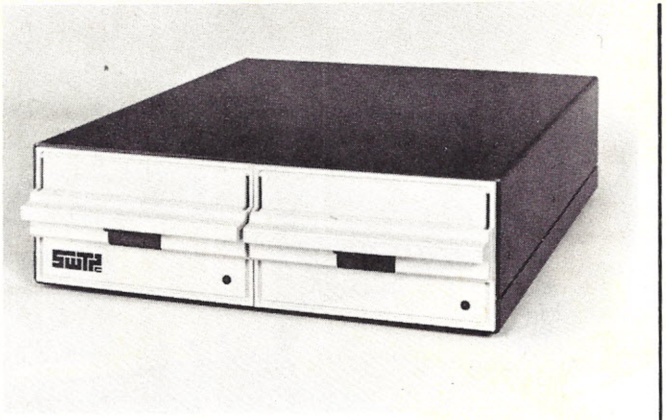
System includes one diskette with Release 3 North Star BASIC and DOS with NORTHSHARE Supervisor and Documentation Package. Price is $48.

Martin Rezmer, Byte Shop of Westminster, 14300 Beach Blvd., Westminster CA 92683.

Prototyping Board Fits DEC and Heath Computers

A new general-purpose prototyping circuit board from Vector Electronic Company permits convenient construction of custom interface circuits for Heath H11 microcomputers and DEC LSI-11, PDP-8 and PDP-11 minicomputers. Form, size and connector-compatible with the DEC "Double-Height, Extended-Length" module, the Vector Model 4607 Plugbord is 8.430 by 5.187 by 0.062 inches.

It has etched contacts spaced to fit the dual 36-pin connectors used in DEC and Heath Computers. Contact terminations are labeled AA1 to AV2 on one set of contacts and BA1 to BV2 on the other set in accordance with DEC nomenclature. To allow unrestricted component placement, the 4607 Plugbord is bare with an array of 0.042 inch diameter holes on 0.1 inch centers. DIP sockets or discrete components may be placed anywhere on the board. Row and column markings are etched into edge strips to insure permanent marking.



SWTP's DMAF1.

The boards are fabricated of blue, epoxy-glass composite material with two-ounce copper cladding. Pin terminations and edge strips are hot-solder plated, while card-edge contacts are gold-flashed nickel plate. The card extractor mounting holes are located to match the outline of standard DEC modules.

In one-to-four quantities, the 4607 Plugbord is priced at $15.95 each; $14.36 in quantities from five to nine; and $12.76 in quantities over 10.

Vector Electronic Company, Inc., 12460 Gladstone Ave., Sylmar CA 91342.

Floppy-Disk System for SWTP 6800

The Southwest Technical Products Corporation DMAF1 is a dual-drive, single-density, double-sided 8 inch floppy-disk system. The hardware consists of an SS-50 bus (SWTP 6800) compatible DMA (direct memory access) controller capable of handling up to four drives, two Cal-Comp 143M double-density-rated disk drives, 5 3/8 x 17 1/8 x 20 1/2 inch aluminum chassis, regulated power supply, drive motor control board, cooling fan, diskette and interfacing cables.

The software supplied is suitable for either the hobbyist or businessman. An 8K BASIC interpreter, with disk file capability and string functions, is also included with the system. Each diskette holds approximately 600,000 bytes (characters) of data, and with two drives you have over one megabyte of data on line.

The system is available in assembled and kit form (the drives themselves are fully assembled) and weighs approximately 45 lbs. Price: $2095 assembled; $2000 kit.

Southwest Technical Products Corp., 219 W. Rhapsody, San Antonio TX 78216.

New H8 Software from Heath

Heath Company has introduced additional software for its H8 personal computer. Extended Benton Harbor BASIC with file capability is a faster, more powerful version of the BASIC software provided with the H8. It includes character strings, more convenience and math functions, dynamic storage allocation and access to a real-time clock. Extended Benton Harbor BASIC requires 12K to 18K of memory and is available in audio cassette or paper tape form. Specify HC-8-13 (mail-order price $20) for the cassette and H8-14 (mail-order price $10) for the paper tape.

New games software for the H8 includes PA-82 Biorhythm, PA-83 Space War and PA-84 Game Set #1, which incorporates Craps, Orbit, Tic-Tac-Toe, Nim, Hexapawn, Hangman, Hmrabi and Derby. Biorhythm runs under Extended Benton Harbor BASIC and requires 16K of RAM. Space War and Game Set #1 require 24K and 8K of RAM, respectively. The games are available in cassette form only and sell for $10 each (mail order, Benton Harbor).

Heath Company, Dept. 350-580, Benton Harbor, MI 49022.

Microchess

Micro-Ware Ltd. announces Microchess 1.5, for the TRS-80 microcomputer, a 4K Z-80 machine-language program utilizing every available byte of user RAM in the TRS-80. The program has been designed to load using the CLOAD command.

Standard algebraic notation is used to describe the moves to the computer. Every move is verified for legality to prevent user error. A simple command allows temporary numbering of the squares to assist in move entry.

The chessboard is displayed using the graphics mode available on the TRS-80. The moving pieces even flash before they move to simulate the gradual narrowing of attention on the moving piece as found in human chess play.

The program has three separate levels of play that will challenge all players from beginners to experienced. Microchess 1.5 is an expanded and improved ver-

Microchess on the TRS-80.

The ABACUS 1 in operation.

sion of Microchess 1.0, which has been available for the 8080 and 6502 microprocessors for over a year. Many user-suggested improvements have been incorporated in this latest release.

Microchess 1.5 is the first in a series of assembler and BASIC programs for the TRS-80: BLOCKADE, ROBOT and LIFE.

Micro-Ware products are available from 27 Firstbrooke Rd., Toronto, Canada M4E 2L2. The price of Microchess 1.5 is $19.95, postage prepaid.

Low-Cost Microprocessor-Based Business System

Computer Products of America, a division of The Computer Mart announces a low-cost, microprocessor-based business computing system that combines accounting functions with word processing. ABACUS 1 is a complete hardware and software package designed to handle basic accounting for small businesses. The system, which includes a Z-80 microprocessor, dual North Star disk system, video display, keyboard and printer plus software, retails for $5995.

According to Computer Products of America, the ABACUS 1 business system is a complete stand-alone system that a small business can put into operation without any additional hardware or software. Programs are written in BASIC.

Functions performed by ABACUS 1 include general-ledger accounting, accounts receivable, accounts payable, inventory, payroll, mailing lists, data entry, sorting and file management. A character-oriented word processing system is available as an option.

The ABACUS 1 features an interactive, double-entry bookkeeping system in which receivables decrease book inventory, payables increase book inventory, and general-ledger accounts are updated automatically with extensive and valid accounting controls. The software package includes 51 programs, with 120 pages of documentation.

Computer Products of America, 633 West Katella Avenue, Orange CA 92667.

CP/M Macro Assembler (MAC)

A new macro assembler, "MAC," is the latest software package offered by Digital Research, Box 579, Pacific Grove CA 93950. MAC operates with the Digital Research standard CP/M Diskette Operating System and implements the recently redefined Intel standard macro facility, while retaining upward compatibility from previous standard assemblers.

Specific features of the new macro processor include conditional assembly (IF, ELSE, ENDIF) with assembly-time expressions (+, –, *, /, MOD, SHL, SHR, AND, OR, XOR, HIGH, LOW, LT, LE, EQ, NE, GE, GT and NUL). Repetition of source statements is provided with indefinite repeat on character (IRPC), indefinite repeat on text (IRP) and numeric repeats (REPT). Parameterized macros are stored using the MACRO statement, which can appear in the mainline source program or be called out from previously defined macro libraries with the MACLIB statement.

Documentation includes the "MAC Macro Assembler Language Manual and Applications Guide," the most complete text available to date on the use of macro facilities for microcomputer software design. Applicable to both the Digital Research and Intel macro assemblers, this 170-page manual contains several complete examples of microcomputer applications.

The diskette containing the macro assembler (machine code only) is available with the documentation for immediate delivery at the price of $70 (diskette order must be accompanied by the purchaser's CP/M serial number). The documentation is available separately for $15 (no serial number required), with the option of later diskette purchase at $60.

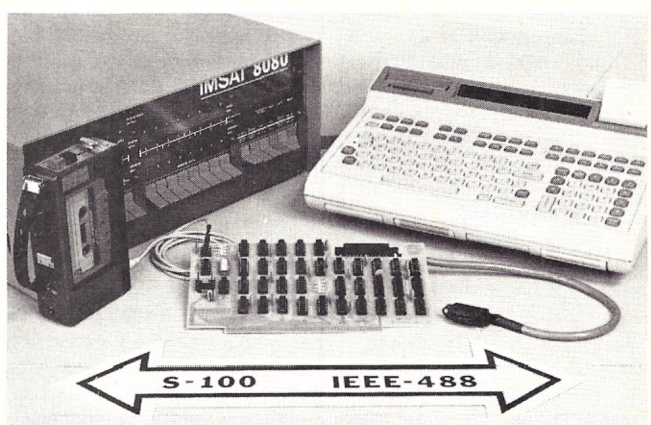
IEEE 488-1975 to S-100 Interface

The P&T-488 interface board provides the broad spectrum of S-100 computers with an interface to the IEEE 488-1975 Standard Digital Interface for Programmable Instrumentation. Using the P&T-488, the computer can function as a talker, listener or controller on the interface bus, allowing intricate instrumentation systems to be configured with S-100 equipment supplying the intelligence.

Software Package 1.0 is distributed with the P&T-488 on a machine-readable cassette tape that can be read with the built-in BITWIGGLER tape interface and a standard audio cassette player. The software is supplied as source code in Intel standard mnemonics, allowing the user to locate the software in the region of memory most suitable to his system. Cable assemblies are included to connect to the 488 bus and the cassette player. Unit price is $250 in kit form and $325 assembled and tested.

Pickles & Trout, PO Box 1206, Goleta CA 93017.

(continued on page 20)



P&T-488 et al.

LETTERS

Camera-Ready Programs —Essential!

It is extremely poor practice to print programs by setting your own type. It is simply asking for trouble, and it would be even if you were, unlike any other magazine in the world, the proud possessor of an infallible proofreader and a stable of mistake-proof authors.

The only guarantee of correctness of a program is to reproduce it photographically from a computer printout, and to be in possession of an affidavit from the submitter that the printout was made just before or after running the program successfully and is unretouched. If the author has no printer, I think it is up to the magazine to have someone else with a printer run it for publication purposes.

My ire was aroused by your April Tic-Tac-Toe article, which was so riddled with obvious errors that I could spot 8 on skimming; God knows how many less glaring mistakes there were. It wouldn't run, anyhow—and no wonder!

Let me suggest in the strongest terms that all printed programs from here on be reproduced from printer output—and that another possible source of confusion and error, excessive concatenation of steps on single lines, be reduced to a minimum. Saving space and printing lots of material is not in the readers' interest if the material lacks accuracy.

A few more experiences like this and I (and maybe lots of other readers) may conclude that *KB*'s output is so unreliable that there's little point in reading it at all!

Richard H. Dorf
New York NY

I couldn't agree with you more about the problems inherent in typesetting programs; and Kilobaud *has published pleas for authors to submit camera-ready programs—either (preferably) printouts or (if necessary) neatly and carefully typed pages. On numerous occasions, we have returned non-camera-ready programs to authors for re-rendering to photographically reproducible quality (you wouldn't believe some of the handwritten scrawls we receive that are called program listings). However, this is not* always *possible. Anything you could do, Richard, to promote authors' submissions of camera-ready programs would be most appreciated.*

While we're on the subject of typographic hassles: trying to edit a single-spaced, all-caps letter to the editor is an unbelievable hassle.—John Barry.

Programs without Articles?

Like most computer hobbyists, I have gone through several phases. I heard about microcomputers; I ordered a subscription to *Kilobaud* (I'm a charter subscriber); I compared the various offerings, wondered if I could afford any; took the plunge and ordered a system; waited what seemed like forever before it arrived; assembled it; figured out how it worked. Now I am the proud owner of an SWTP system with a printer and adequate (20K) memory. And now, from talking to other hobbyists, I am in the final phase, which is an endless loop: the program freak.

Wayne Green once wrote that the uses of a micro are limited only by one's imagination. True, but unfortunately I must be cursed by a fairly limited imagination. I have written several useful and fun programs, and have no doubt that the micro was a good investment—but I always tear the jacket off of *Kilobaud* every month looking for new programs. Sometimes, the simplest are the most interesting because they give me ideas for new uses. As an example, the program you published last year allowing the micro to print out "The Twelve Days of Christmas" was both unusual and interesting. I learned quite a lot about handling string variables from reading the listing for that program.

I would love to share my programs with others, and I suspect that there are a lot of other hobbyists who would gladly do the same. I like the idea of being paid for them, but that isn't the primary reason I haven't submitted any to you for consideration. The reason—and I think I share this with others—is that I have neither the time nor the ability to write the obligatory article that seems to accompany every program listing.

So, I suggest that you consider publishing BASIC programs—without long articles, and without big pay to the authors—in a special part of the magazine. You might develop a submittal sheet with the program listing to indicate the program limitations, the type of BASIC the author used, the memory and other requirements. The sheet should be easy to fill out and available on request.

I think that accepting BASIC programs in this fashion (rather than requiring the author to write both the program *and* an article) would smoke a lot of useful programs and ideas out of the woodwork; and I also think it would make *Kilobaud* a more useful magazine.

Stephen L. Carter
Rifle CO

Fine. We've had requests for this before, and we're willing to give it a try. The remarks at the beginning of the program must *do an adequate job of explaining how to load and operate the program, what kind of BASIC is being used, memory requirements, any tips known by the author on converting over to another BASIC, and whatever else might be of value. Go to it!—John.*

What's Your Sine?

There is an article in the March 1978 issue of *Kilobaud* entitled "Number-Crunching Time." The author (Mel Baker) uses successive approximations to obtain the value of x for the equation:

$$\sin(x+2) - e^x + x^2 + 3.701 = 0.$$

Even with pencil and paper, I cannot obtain the results reported by the author.

For instance, if $x = 0$, then $\sin(2) = .0348995$, and $e^\circ = 1$. The result for the equation is 2.736, and the author reports 3.610. How come?

I am no expert on computers, but on the big machines I learned long ago that failure to understand the subtlties of "computer mathematics" can sometimes be disastrous. The source of the problems seems to be the inevitable truncations that arise in various ways.

How about getting someone who really understands these problems to prepare an article about them?

Robert G. Hoffmann, PhD
Miami FL

Mel Baker's Reply:

I am sure that if he were asked to add two feet to three inches, Dr. Hoffman would readily recognize that he would have to either change feet to inches or inches to feet before calculating.

He has been caught by a similar, but more subtle, trap in this problem. If x is measured in degrees, $\sin(x+2)$ will be a real number, while x^2 will have the strange units degrees squared. If this were the case we would be adding unlike units.

On the other hand, if x is a real number there is no problem since all four terms will be reals.

If Dr. Hoffman will multiply 2 by 180° and divide by pi, the value of x will be converted to degrees and the sine will come out .9093, which will give the result shown in the article. Another way would be to find the sine of 2 using radians.

It is important that you not think of radians as units, however. Again x^2 would be radians squared, etc. Radians are unitless in this sense.

Hope this clears it up.

Mel Baker
Associate Prof. of Math.
Union College
Lincoln NE

PROM Part Numbers— Watch Out!

I just came across a very untidy situation involving PROM numbers that may be of interest to a great number of readers. We are all familiar by now with the advantages of the 2716 EPROM; however, everyone should be aware that while the Intel version (and possibly others) requires only a single +5 volt supply, Texas Instruments makes a similar device with identical memory organization, number TMS2716, that requires ±5 and +12 volt supplies. This part is designed to be pin compatible with the older 2708s. TI has another part labeled TMS2516, which is pin compatible with the Intel type 2716, requiring only a +5 volt supply. While this is assuredly somewhat bizarre, there seems to

be argument among the subject companies as to who claimed the name 2716 first; it is definitely one that many will be in need to be aware of. Perhaps you won't get caught in this mess inextricably.

David Marke
Chief, Research and Development
Solar Dynamics, Ltd.
Austin TX

A Better Mailing-List Routine

I devoured your latest issue from cover to cover and found it to be low in fat and high in protein as usual.

Tom Doyle's article, "5 Minutes or 5 Hours," was particularly informative. Having just completed entering the mailing-list program from the preceding article, "Strings and Things," I was very tempted to replace the bubble sort in that program with the Shell-Metzner sort as described by Tom Doyle.

I did just that and added an option to have the alphabetized list in first- or last-name order. It was a lot of fun to do, and I now have an effective and fast mail-sort program, which is enclosed.

Keep up the good work and keep that software rollin'.

Plato J. Grivas, M.D.
Alamo CA

```
10 REM THIS IS A MAIL-LIST WITH S/M SORT AND CHOICE OF LAST NAME
20 REM THIS IS A MAILING LIST (HP STYLE)
30 READ N9
40 DIM N$(N9*30),F$(60),F1$(30),F2$(30),A$(30)
50 REM USE FUNCTIONS FOR PSEUDOMATRIX OF STRINGS
60 DEFFNL(X)=(X-1)*30+1\DEF FNH(X)=X*30
70 DEF FNA$(A$)
80 IF LEN(A$)>=30 THEN RETURN A$
90 A$=A$+" "\GOTO 80\FNEND
100 REM IN NAMES
110 !"****NAMES****"\!
120 N$=""\REM CLEAR MATRIX
130 FOR I=1 TO N9
140 READ F$
150 !F$
160 F$=F$+"$"\REM MARK END OF NAME FOR REVERSE ROUTINE
170 F$=FNA$(F$)\REM FILL NAME TO 30 CHARS
180 N$=N$+F$
190 NEXT I
200 REM DATA
210 LINE 80
220 DATA 10
230 DATA"SALLY JONES","SAM SMITH","JOE SMITH","TIM CAMPBELL","ED HILL"
240 DATA"STEVE MOODY","ROGER HEAD","SHIRLEY JONES","ISAAC DEAR","RICH KING"
250 REM RE-ORDER LAST NAME FIRST
260 FOR N1=1 TO N9
270 F$=N$(FNL(N1),FNH(N1))
280 C=1
290 REM LOOP UNTIL END MARK IS FOUND
300 IF F$(C,C)="$"THEN 340
310 IF F$(C,C)=" "THEN S=C
320 C=C+1
330 GOTO 300
340 REM REVERSE FIRST & LAST NAMES
350 F1$=F$(1,S-1) \REM FIRST NAME
360 F2$=F$(S+1,C-1) \REM LAST NAME
370 F$=F2$+", "+F1$
380 REM PUT BACK IN MATRIX(NOTE FULL 30 CHARS SO NO LEFT OVERS)
390 N$(FNL(N1),FNH(N1))=FNA$(F$)
400 NEXT N1
410 !"COMPUTER WORKING",
420 REM SET UP ARRAY
430 N=N9\J=N
440 FOR I=I TO N
450 F$=N$(FNL(J),FNH(J))
460 J=J-1
470 NEXT
480 REM SHELL-METZNER SORT
490 M=N9
500 M=INT(M/2)\IF M=0 THEN 670
510 J=1\K=N-M
520 I=J
530 L=I+M
540 C0=C0+1
550 IF N$(FNL(I),FNH(I))<N$(FNL(L),FNH(L)) THEN 640
560 REM SWAP
570 F$=N$(FNL(I),FNH(I))
580 N$(FNL(I),FNH(I))=N$(FNL(L),FNH(L))
590 N$(FNL(L),FNH(L))=F$
600 S0=S0+1
610 I=I-M
620 IF I<1 THEN 640
630 GOTO 530
640 J=J+1
650 IF J>K THEN 500
660 GOTO 520
670 !\!\INPUT"DO YOU WANT LAST NAMES FIRST? (Y OR N) ",Z$
680 IF Z$<>"Y" THEN X=1
690 REM
700 REM PRINT SORTED LIST
710 !\!\!"**** SORTED NAMES ****"\!\!
720 FOR N1=1 TO N9
730 F$=N$(FNL(N1),FNH(N1))
740 IF X=0 THEN 760
750 GOSUB 790
760 !F$
770 NEXT N1
780 END
790 REM RE-ORDER FIRST NAME FIRST
800 F$=N$(FNL(N1),FNH(N1))
810 C=1
820 IF F$(C,C)=" "THEN 860
830 IF F$(C,C)=","THEN S=C
840 C=C+1
850 GOTO 820
860 F1$=F$(1,S-1) \REM LAST NAME
870 F2$=F$(S+1,C-1) \REM FIRST NAME
880 F$=F2$+" "+F1$
890 N$(FNL(N1),FNH(N1))=FNA$(F$)
900 RETURN
READY
```

Program listing.

Mini Micro Mart . . . Again!

I am sure you are tired of hearing about the troubles that various computer hobbyists have been having with Mini Micro Mart, but as it is partially your fault in this case, please read on.

Let me explain your involvement in my case. I had been warned not to do any business with this company many times. I found, however, that they had some circuit boards I needed and at a rather good price, so when I read the two letters concerning Mini Micro Mart in issue No. 10 of *Kilobaud* ("Around the Industry") I thought I would give them a chance. After all, no one could actually give service as poor as all the stories seemed to indicate and stay in business very long, right? WRONG!

The only thing I ever received from MMM was a partial shipment and a total brush-off. After four months of writing letters trying to get some response from them, all I have is two canceled checks and a return receipt from a certified letter I sent them. I am now in the process of filing a formal complaint with the Federal Trade Commission charging gross violations of mail-order Rule 435.

You would be doing a service to all honest companies as well as all your readers to warn against doing any business with Mini Micro Mart.

Robert J. Retelle
Ypsilanti MI

I've been trying to tell people about MMM for 2½ years now. I'll continue to do so. I ran those two letters in Kilobaud *No. 10 because I figure everyone is entitled to defend himself after being chastized in print. So much for the "defense." The letters of*

complaint continue to roll in. What more can I say? Be sure to read John Lehman and Ray Graham's article on MMM in issue No. 18 of Kilobaud.*—John.*

Smythe Topless?

I subscribed to *Kilobaud* for the information content. . . .

If you *must* do personal pieces and pictures, please try to have Helmers clothed and Smythe topless.

In any case, please continue sending *Kilobaud* in a plain brown wrapper as before. Thank you.

Roy O'Brien
South Bound Brook NJ

I've got a much better idea . . . why don't we make every effort to concentrate on personal computing.—John.

"Deflection" for the Poly-88

I have modified the game of Deflection (February 1978 *Kilobaud*) to run on a Poly-88 computer. Other Poly users should find the program interesting and workable if they make the indicated changes.

The Poly assemblers use a slightly different notation from the one in the article, so it is necessary to make the changes in Fig. 1 throughout the program.

Change the EQUates at the beginning of the program as in Fig. 2.

VDMPORT is not needed in the Poly program. The EQUates for the directions are OK as is.

For the remainder of the changes, I will refer to the addresses printed in the article listing as if they were line numbers (see Fig. 3).

And finally, delete line 01E4 since the Poly operating system will have already set the stack.

The TARGET and RUNNER symbols are not the same as in the article, since the Poly character set does not contain those exact characters. If you don't like the characters I've chosen, you can pick your own by changing the EQUates for TARGET and RUNNER.

I developed the changes to this program using the G02 version of the Poly-88 Editor/Assembler. This version is a lot handier to work with than the old 4.6 version of the assembler. The new version gives you a lot of character-editing capability. If you don't have a G02 yet, I suggest you get a demonstration of its capabilities from your computer store or from another Poly owner.

David Larry Johnson
Prince George VA

Article	*For Poly Change To*	
DE	D	
BC	B	
HL	H	
x‘nnn’	OnnnH	(where "n" stands for a hex character)

Fig. 1.

Symbol	*Article*	*For Poly Change To*	
STATUS	x‘06’	0F9H	
STROBE	x‘02’	1H	(or just 1)
DATA	x‘05’	0F8H	
VDMRAM	x‘CC00’	0F800H	
VDMPAGE	x‘CC’	0F8H	
TARGET	x‘05’	087H	
RUNNER	x‘07’	098H	
USLSH	x‘2F’	0AFH	
DSLSH	x‘5C’	0DCH	
CLSCRN	EQU 392H	(add this to use the Poly monitor screen clear routine instead of the article clear routine)	

Fig. 2.

"Line"	*Article*	*For Poly Change To*
0050	GSTART, etc.	DI
0053	CALL CLSCRN	GSTART CALL CLSCRN
0064	x‘20’	7FH
00D3	x‘20’	7FH
00F8	x‘C0’	0F0H
0102	x‘20’	7FH
0125	x‘20’	7FH
Add DI between lines 0188 and 018B		
0195	ANI x‘7F’	ORI 80H
Delete lines 019F through 01AE		
Replace lines 01CC through 01DE with:		
WIN DB	0D7H,0C9H,0CEH,0C5H,0D2H,7FH	
LOSE DB	0CEH,0CFH,7FH,0D4H,0C9H,0CDH,0C5H	
BLNK DB	7FH,7FH,7FH,7FH,7FH,7FH,7HF	

Fig. 3

TROUBLE-SHOOTERS' CORNER

(from page 7)

bench and taken perhaps ten times as long to do it. It's not that I'm stupid or that he was a genius. It was just an unforgettable example of the trade-off between knowledge and test equipment. The more knowledge you have, the less equipment you need —and vice versa. In fact, the evolution of a new development can almost be measured by the amount of test gear hooked up to it.

In the early stages, when knowledge is being developed, the prototype is often buried in a maze of test leads and E-Z-hooks. When it gets to the production stage, a simple logic probe may be all that's required, if that. Obviously, I'm not proposing that you throw out your test equipment—only that you put it in its proper perspective.

Let's say you're betting on a hare-and-tortoise-type race between two competent technicians (of equal intelligence) who have to find a bug in your microcomputer in the least possible time. One has a complete laboratory with all the latest *things*. The other has only a $25 logic probe and a little book of numbers—telephone numbers of the right people. Whom would you bet on?

Step 2

I started by suggesting that your first step was to take an inventory of your assets. A very important asset is knowing how to utilize other assets to solve your problem. The next step is to define the problem. The last step is, of course, to fix it. In practice, this is usually the easiest part (if it's at all possible), providing you did a thorough job on step 2.

Most of the succeeding information in this series will center around *defining* the problem. Most of the time, "fixing" it will consist of changing a chip or a program instruction. This can usually be done in a few minutes, but finding it can drive you up a wall.

Hardware or Software?

Microcomputers are unique in their ability to interchange hardware and software. Software can be used to "box in" hardware problems, and vice versa. For instance, I'm sure we will get into software routines for finding bad memory chips (hardware). This same trade-off can often provide a better solution (cornerstone three) than the original design.

There are several common problems that could be solved using either hardware or software. Often the best solution is a mixture. As an example, there are now chips available that can do arithmetic functions faster than the traditional software routines. On the other hand, most of the 6502-based systems handle TTY and audio/cassette serial interfacing with general-purpose PIAs and software timing, instead of the usual ACIA large scale integrated circuits (LSI). Vertical and longitudinal parity can be generated by either hardware or software, thus becoming a speed-vs-cost decision.

When it comes to defining problems and providing solutions, the trade-offs between hardware and software should provide some interesting approaches to troubleshooting. If you have either interesting solutions or common problems from which others can benefit, drop a line to Troubleshooters' Corner, c/o *Kilobaud*. The topics dis-

cussed in this column will be determined by your feedback. I may not have the answer, but I do have a lot of friends. So until next month, remember: Like a growing puppy, your computer thrives on friendship—synergistic, synectic friendship.

KB CLUB CALENDAR

(from page 10)

Singapore

Richard Bash would like to hear from computer hobbyists in Southeast Asia who are interested in "pooling knowledge, swapping software, etc. Are there any microcomputers out there whose owners want to start a club?" Write to Richard at 5 Jalan Layang-Layang, Singapore 21, Republic of Singapore.

San Francisco CA

The Apple Core has sprung to life in 'Frisco, according to the group's organizer, Scott Kamins. To quote their April newsletter (I'm a bit late), "To qualify as a member of The Apple Core you must own or regularly use an Apple in any memory configuration. You must also pay dues. . . ." Anyone who can meet these outrageous requirements can write to The Apple Core, Box 4816, San Francisco CA 94101.

This column is available for you to report on your club's activities such as regular meeting schedules, special events or programs, swap meets or any endeavor that will be of interest to your fellow hobbyists. If your announcement contains timely information, please send it at least two months prior to the date or dates mentioned in the announcement.

Kilobaud Club Calendar
c/o Steve Fuller
334 Sterling St. Unit A-3
West Boylston MA 01583

CORRECTIONS

Regarding Mike Kop's article, "Do-It-Yourself Time-sharing," in the May issue, Jim Howell writes: "It is not required to have a very short pulse width when using the 6800's nonmaskable interrupt (NMI). This is because NMI is edge-triggered. A high-to-low transition of NMI causes an interrupt. After this, NMI can stay low as long as you want it to without causing any more interrupts. (This is different from the IRQ interrupt, which is level-triggered.)"

Gerry Wheeler also wrote with similar comments.

KB CALENDAR

Tidewater Radio Conventions, Inc., announces its Third Annual Hamfest, Flea Market and Computerfest to be held on September 23 and 24 1978 at the Norfolk Cultural and Convention Center (SCOPE). This is an approved A.R.R.L. Function. Over 65,000 square feet of indoor, air conditioned facilities for commercial exhibits, meetings, and flea market are available.

Videcom's VMR-120.

NEW PRODUCTS

(from page 16)

Video Monitor/Receiver

Videcom Division of General Technical Products announces their new model VMR-120 12-inch diagonal monochrome Video Monitor/Receiver. The chassis is 100 percent solid-state design and is power-transformer-isolated from the ac line. The VMR-120 is ideally suited for industrial, security, studio or computer display applications, with separate UHF connections for video IN and THRU, RCA connector for audio IN and a termination switch. Horizontal resolution is rated at 550 lines center. The VMR-120 is also available as a monitor only.

Model VMR-120A has the additional features of an 8 pin VTR connector and separate UHF and RCA connectors for TV video and audio out for off-the-air recording. Power requirements are 117 V, 60 Hz, 30 Watts, or 12 to 16 V dc, 14 Watts. Weight is 16 lbs. and the units are equipped with a carrying handle. User prices are $199 for the VMR-120 and $215 for the VMR-120A. A 19-inch monochrome Receiver/Monitor for VTR or security use is available, with an end user price of $350.

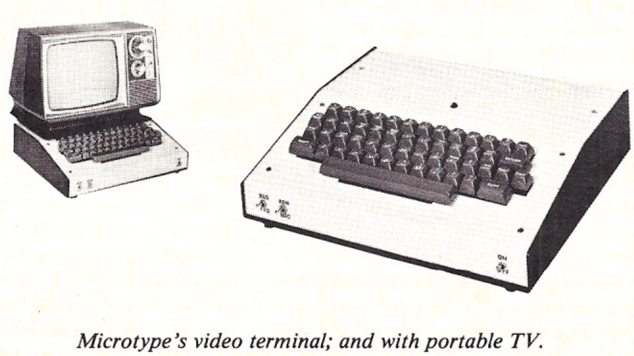
Video Marketing, Inc., 328 Maple Ave., Horsham PA 19044.

Video Terminal by Microtype

Microtype's new video terminal is a low-cost, quiet alternative to the Teletype Model 33ASR. It will display data on any black and white transformer-powered TV set modified to include a video input jack. The terminal features a 16 line by 64 character format, auto scroll, auto line feed and auto return, selectable baud rate (110 and 300) and quartz crystal timing.

The compact terminal measures only 3 x 13 x 13 inches. All controls are conveniently mounted on the front panel of the heavy-gauge metal case. Both RS-232C and 20 mA current loop interfaces are provided. The fully assembled video terminal is priced at $350. A modified Zenith 9 inch TV can be supplied with the video terminal for an additional $100. It can still be used as a TV receiver.

Microtype, Inc., 404 E. Park, Plano IL 60545.



Microtype's video terminal; and with portable TV.

DOCUFORM: A Word-Processing System for Everyone!

This comprehensive program can greatly streamline the amount of time and money you may now spend in processing written material.

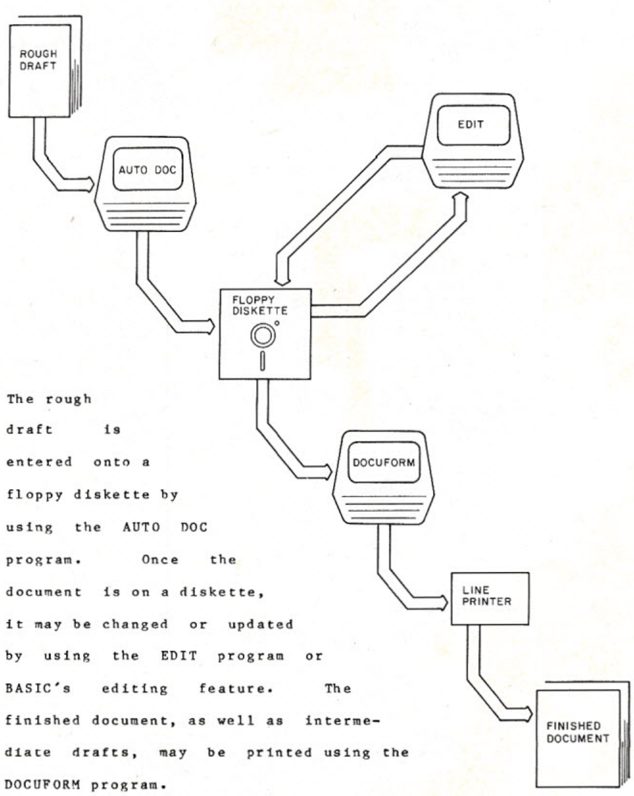
Donald L. Fitchhorn
Business Programmer
Albuquerque Computing Services
MSD/PCC

Does your small business have mountains of correspondence, reports and proposals, but insufficient capital to purchase a modern word-processing system? The DOCUFORM programs can turn your microcomputer into a powerful word-processing system and minimize your typing, retyping and paper shuffling.

Simply stated, word processing is reproducing written material in the most expedient, "painless" manner possible. The basic principle in word processing is to make the typist's job as easy as possible.

Imagine the typist who is asked to produce a rough draft of a 200-page report. This person realizes that the report is bound to be sent back with revisions in a few days and will have to be typed all over again. Then, of course, the revision will be revised and will have to be retyped. Since typists are usually paid by the hour, they normally don't complain about this procedure.

The waste here is obvious. Most of the text in the rough draft will remain the same in successive revisions, and receive only minor modifications. This waste can be eliminated by storing the rough draft and making changes to the stored text. Each new revision is then just a matter of making neces-



Flowchart 1. DOCUFORM system use.

```
The DOCUFORM System has many advanced capabilities which
make the typist's
job easier.  But in order to
understand DOCUFORM's advanced capabilities, the more basic
functions must be
explained.

Can print as:
The DOCUFORM System has many advanced capabilities which
make the typist's job easier.  But in order to understand
DOCUFORM's advanced capabilities, the more basic functions
must be explained.
```

Example 1. Text.

```
The DOCUFORM System has many advanced capabilities which
make the typist's
job easier.  But before you can be expected to
understand DOCUFORM's advanced capabilities, the more basic
functions must be
explained in detail.

Can print as:
The DOCUFORM System has many advanced capabilities which
make the typist's job easier.  But before you can be
expected to understand DOCUFORM's advanced capabilities, the
more basic functions must be explained in detail.
```

Example 2. Edited text.

sary changes and printing the new version. The DOCUFORM Word-Processing System utilizes this time-efficient procedure and incorporates several other features that make the typist's job easier. See Flowchart 1 for a visual representation of the way the DOCUFORM system works.

Basic Features

Line filling. This is the most basic feature provided by DOCUFORM. Line filling is the process of adding words to a line until one more word will make the line longer than the specified margin. This word, and the words following it, are saved for printing on the next line. This allows the saved version of the text to have lines of different lengths from that which is printed (Example 1).

To make changes to a saved text file, all the typist has to do is remove (or add) the lines (or words) necessary for the revision. When the new version of the text file is printed, it will appear to have been completely retyped. In Example 2 only the third and sixth lines had to be changed from Example 1 in order to get the new paragraph.

Hyphenation. At times, however, the lines produced by line filling are shorter than desired. This is caused by a long word being put on the next line because this word would make the current line too long. This problem can be solved by hyphenating the word at a place that will allow the most letters to remain on the current line. DOCUFORM has three different methods of hyphenating.

1. Embedded hyphens.
2. Dictionary lookup.
3. Operator query.

If the word contains a hyphen (as in the compound self-enhancing), DOCUFORM will divide the word after the hyphen if the letters before the hyphen (plus the hyphen, of course) will fit on the current line. For example, if self- fits on the current line, DOCUFORM will split the word self-enhancing. Otherwise, self-enhancing will appear on the next line.

When DOCUFORM comes across a candidate for hyphenation that has no embedded hyphen, it first looks in a file called DICTNARY. This file contains words and their correct points of hyphenation. If the word is found, it is divided at the most appropriate of the letter pairs designated as hyphenation points by DICTNARY.

If the word is not found, the typist is asked for the hyphenation points. (The typist is encouraged to look the word up in a dictionary because the hyphenation points entered at that time will be automatically stored in DICTNARY for future use.) DOCUFORM then proceeds as if it had found the word in DICTNARY.

Justification. Once a line has been filled, it is ready to be printed and the material will appear as though it were typed. But in some applications (such as newspapers, magazines and books) it is desirable for the text to appear with the right margin straight like the left margin. This is called *justification*. DOCUFORM can produce this effect in any of three different ways:

Mode 1. Adding blanks.
Mode 2. Spreading blanks.
Mode 3. Spreading space between characters.

```
10 'THIS IS A PROGRAM FILE
20 FOR I = 1 TO 100000
30 PRINT I;RND(I)*1000;I*RND(I)
40 ' THIS PRINTS SOME NUMBERS
50 NEXT
60 END

10 'This is a program file
20 'here is the
30 'text of the
40 'document !
50 'this is the end.
```

Example 6. Program/Document files.

The DOCUFORM System has many advanced capabilities which make the typist's job easier. But in order to understand DOCUFORM's advanced capabilities, the more basic functions must be explained.

Example 3. Justification mode 1.

The DOCUFORM System has many advanced capabilities which make the typist's job easier. But in order to understand DOCUFORM's advanced capabilities, the more basic functions must be explained.

Example 4. Justification mode 2.

The DOCUFORM System has many advanced capabilities which make the typist's job easier. But in order to understand DOCUFORM's advanced capabilities, the more basic functions must be explained.

Example 5. Justification mode 3.

COMMAND LIST	FORMAT
MARGIN CONTROL	
.BM - Bottom Margin	.BMn
.LM - Left Margin	.LMn
.LS - Left Side	.LSn{,n,n,n,...}
.MA - Margin Adjust	.MAm
.NM - New Margin	.NMn,n (left,right)
.OM - Old Margin	.OM
.OV - OVer	.OVn
.PL - Page Length	.PLn
.RM - Right Margin	.RMn
.RS - Right Side	.RSn{,n,n,n,...}
.TM - Top Margin	.TMn
FORMATING	
.BL - Blank Line	.BL{n}
.CE - CEnter line	.CEd
.FL - FiLl	.FL
.IT - IndenT	.ITm
.JU - JUstify	.JUn (1,2,or3)
.LF - Line Format	.LFd
.NF - No Fill	.NF
.NJ - No Justify	.NJ
.SP - SPacing	.SPn
.SY - SYllabication	.SYn (0or1)
CARRIAGE CONTROL	
.HD - HeaDer	.HDd
.PA - PAge	.PA{n}
.TP - Test Page	.TPn
.WA - WAit	.WA
.WT - Wait at Top	.WTn (0or1)
SPECIAL FEATURES	
.CH - CHange character	.CHcn
.CM - CoMment	.CM{?}d
.CO - COntents	.COd
.DF - Data File	.DF{f}
.EM - Execute Macro	.EMo
.IN - INdex	.IN
.NI - No Index	.NI
.NX - NeXt file	.NXf
.PK - PoKe	.PKn,n
.RP - RePeat file	.RP
.SH - Standard Header	.SHof
.SM - Set Macro	.SMo.a{.a.a.a....}
.ST - STandard paragraph	.STf
?? - input from data file	
^ - precedes word for index	

FORMAT ABBREVIATION EXPLANATIONS

a - command
c - character
d - data
f - file name
n - positive number
m - positive or negative number
o - digit 0-9
() - see text
{} - optional

Table 1. Command list with formats.

Example 7. Sample input files with output.

```
10 '.CM  THIS IS A DOCUFORM FILE WHICH PRINTS ITSELF 4 TIMES
20 '.CM
30 '.CM  SET LEFT AND RIGHT MARGINS,
40 '.CM  POKE QUME DRIVER FOR APPROXIMATELY 12 PITCH
50 '.CM
60 '.LM10.RM80.SY1.PK15500,11.PK15517,11.PK15599,14
65 '.CM
70 '.CH#32.CM  CHANGE # TO BLANK
80 '.CM
90 '.CM  SET JUSTIFICATION TO MODE #3, SPACING TO 2, PAGE LENGTH TO 75,
100 '.CM  AND OPEN A DATAFILE NAMED CHANGABS
105 '.CM
110 '.JU3.SP2.PL75.DFCHANGABS
115 '.CM
120 '.CM  PICK UP A COMMAND FROM CHANGABS
130 '??
140 'One of the most helpful tools to be introduced
150 'in recent years is computerized text formating. There
160 'are hundreds of different text formating systems
170 '(a series of computer programs) being sold
180 'today, but none of them appear to be designed for the
190 'businessman with a small budget.
200 'A small business rarely has the kind of money that it takes
210 'to buy or rent the machines that these systems use.  New developments
220 'in micro-computers, however, have brought machine costs down by a factor
230 'of a hundred without hampering their ability to handle complex tasks.
240 'The DOCUFORM system was developed with this in mind.
250 'It runs on the ALTAIR 8800,
260 'is easy to use,
270 'and packs more power and versatility than some systems
280 'costing thousands of dollars.  DOCUFORM includes: three modes of
290 'justification, automatic index, automatic table-of-contents, data
300 'insert, standard paragraphs, and automatic hyphenation.  To make
310 'the system complete, it comes with a text editor, and a pre-processor
320 'that automatically inserts DOCUFORM commands while you type.
330 'The equipment needed to utilize the full capability of the DOCUFORM
340 'system can be purchased for less than $12,000.
350 'Finding another system that does this much for the same price
360 'will not be easy.
365 '.CM
370 '.CH#35.CM  CHANGE # BACK TO #
375 '.CM
380 '.CM  PERFORM HOF, POKE QUME BACK TO NORMAL, REPEAT THE FILE
385 '.CM
390 '.PA.PK15500,12.PK15517,12.PK15599,16.RP
```

```
            THIS IS CHANGABS; THE DATA FILE
10 '.STNORMAL
20 '.STCIRCLE
30 '.STNUMB-ONE
40 '.STTREE

10 '.CM?  THIS IS THE NORMAL FILE
20 '.CEThis is how the file will look normally
30 '.BL3

100 '.CM?  THIS IS THE CIRCLE FILE
110 '.CEThis is how the file will look as a circle
120 '.BL2
130 '.CM
140 '.CM  SET UP THE LEFT SIDE FOR MOVING MARGINS
150 '.CM
160 '.LS28,23,20,17,15,14,12,10,9,9,8,7,6,6,5,5,5,4,4,4,4,4,4,4,4,4,4,4,5
,5,5,6,6,7,8,9,10,10,12,14,15,17,20,23,28
170 '.CM
180 '.CM  SET UP THE RIGHT SIDE FOR MOVING MARGINS
190 '.CM
200 '.RS48,53,57,59,61,63,65,67,68,69,70,71,71,72,72,73,73,74,74,74,74,74
,74,74,74,74,74,73,73,72,72,71,71,70,69,68,67,65,63,61,59,57,53,48

100 '.CM?  THIS IS THE NUMBER ONE FILE
110 '.CEThis is how the file will look as a number one
120 '.BL2.SP1
130 '.CM
140 '.CM  SET UP THE LEFT SIDE FOR MOVING MARGINS
150 '.CM
160 '.LS25,24,23,21,19,17,14,9,9,9,9,9,9,9,9,9,20
170 '.CM
180 '.CM  SET UP THE RIGHT SIDE FOR MOVING MARGINS
190 '.CM
200 '.RS50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50

100 '.CM?  THIS IS THE TREE FILE
110 '.CEThis is how the file will look as a tree
120 '.BL2.SP1
130 '.CM
140 '.CM  SET UP A LEFT SIDE FOR MOVING MARGINS
150 '.CM
160 '.LS41,40,38,36,33,30,26,36,35,33,31,28,25,21,31,30,28,26,23,20,16
,26,25,23,21,18,15,11,21,20,18,16,13,10,35
170 '.CM
180 '.CM  SET UP A RIGHT SIDE FOR MOVING MARGINS
190 '.CM
200 '.RS45,46,48,50,53,56,60,50,51,53,55,58,61,65,55,56,58,60,63,66,70
,60,61,63,65,68,71,75,65,66,68,70,73,76,51
```

```
          This is how the file will look as a number one


                    One of the most helpful
                   tools to be introduced in
                  recent years is computer-
                ized text formating. There
              are hundreds of different text
           formating systems (a series of
        computer programs) being sold today,
    but none of them appear to be designed
    for the businessman with a small budget.
    A small business rarely has the kind of
    money that it takes to buy or rent the
    machines that these systems use.  New
    developments in micro-computers, however,
    have brought machine costs down by a
    factor of a hundred without hampering
    their ability to handle complex tasks.
              The DOCUFORM system was devel-
              oped with this in mind. It
              runs on the ALTAIR 8800, is
              easy to use, and packs more
              power and versatility than
              some systems costing thousands
              of dollars.  DOCUFORM
              includes: three modes of jus-
              tification, automatic index,
              automatic table-of-contents,
              data insert, standard para-
              graphs, and automatic
              hyphenation.  To make the
              system complete, it comes with
              a text editor, and a pre-
              processor that automatically
              inserts DOCUFORM commands
              while you type.  The equipment
              needed to utilize the full
              capability of the DOCUFORM
              system can be purchased for
              less than $12,000.  Finding
              another system that does this
              much for the same price will
              not be easy.
```

```
              This is how the file will look as a tree


                               One
                              of the
                             m o s t
                           helpful tools
                        to be introduced in
                      recent years is computer-
                   ized text formating. There are
                           hundreds of
                          different text
                        formating systems (a
                       series of computer
                    programs) being sold today,
                   but none of them appear to be
              designed for the businessman with a small
                        budget.   A small
                       business rarely has the
                    kind of money that it takes to
                  buy or rent the machines that
               these systems use.  New developments in
            micro-computers, however, have brought machine
          costs down by a factor of a hundred without hampering
                  their ability to handle complex
                 tasks.  The DOCUFORM system was
              developed with this in mind. It runs on
           the ALTAIR 8800, is easy to use, and packs
        more power and versatility than some systems
      costing thousands of dollars.  DOCUFORM includes: three
  modes of justification, automatic index, automatic table-of-
          contents, data insert, standard paragraphs,
        and automatic hyphenation.  To make the system
       complete, it comes with a text editor, and a pre-
    processor that automatically inserts DOCUFORM commands
  while you type.  The equipment needed to utilize the full
capability of the DOCUFORM system can be purchased for less than
                      $ 1 2 , 0 0 0 .
                      Finding another
                      system that does
                      this much for
                      the same price
                      will not be
                      easy.
```

Adding blanks (see Example 3) is the most basic method used to justify lines. DOCUFORM first determines how many characters are needed to expand the line to the specified length. The program then adds extra blanks into the line by inserting them one at a time wherever a blank occurs in the original line. By alternating between starting from the right or the left end of the line, DOCUFORM makes the added blanks less noticeable. Although this method takes the longest to

This is how the file will look as a circle

One of the most
helpful tools to be introduced in
recent years is computerized text formating.
There are hundreds of different text formating
systems (a series of computer programs) being sold today,
but none of them appear to be designed for the businessman
with a small budget. A small business rarely has the kind of
money that it takes to buy or rent the machines that these systems
use. New developments in micro-computers, however, have brought
machine costs down by a factor of a hundred without hampering their
ability to handle complex tasks. The DOCUFORM system was developed
with this in mind. It runs on the ALTAIR 8800, is easy to use, and
packs more power and versatility than some systems costing thousands
of dollars. DOCUFORM includes: three modes of justification, auto-
matic index, automatic table-of-contents, data insert, standard
paragraphs, and automatic hyphenation. To make the system
complete, it comes with a text editor, and a pre-processor that
automatically inserts DOCUFORM commands while you type. The
equipment needed to utilize the full capability of the
DOCUFORM system can be purchased for less than
$12,000. Finding another system that does
this much for the same price
will not be easy.

This is how the file will look normally

One of the most helpful tools to be introduced in recent years is computerized text formating. There are hundreds of different text formating systems (a series of computer programs) being sold today, but none of them appear to be designed for the businessman with a small budget. A small business rarely has the kind of money that it takes to buy or rent the machines that these systems use. New developments in micro-computers, however, have brought machine costs down by a factor of a hundred without hampering their ability to handle complex tasks. The DOCUFORM system was developed with this in mind. It runs on the ALTAIR 8800, is easy to use, and packs more power and versatility than some systems costing thousands of dollars. DOCUFORM includes: three modes of justification, automatic index, automatic table-of-contents, data insert, standard paragraphs, and automatic hyphenation. To make the system complete, it comes with a text editor, and a pre-processor that automatically inserts DOCUFORM commands while you type. The equipment needed to utilize the full capability of the DOCUFORM system can be purchased for less than $12,000. Finding another system that does this much for the same price will not be easy.

perform, it will work with any type of printer or terminal.

Spreading blanks (see Example 4) is more commonly used and faster than adding blanks. It can be seen in many books and magazines. This method of justification (as well as spreading spaces between characters) requires a special printer to produce its effect. The printer must be able to change the width of the space between characters as it is printing. DOCUFORM spreads blanks in much the same manner as it adds blanks. Once it knows how much extra space is needed to fill out a line, it instructs the printer to print each blank just wide enough to make the line the desired length.

Spreading spaces between characters (see Example 5), used mostly in newspapers, is produced by altering the width of the space between each character in the line while it is being printed. This method is the fastest of the three.

DOCUFORM's Capabilities

Line filling and justification are the main aspects of computerized text formatting. However, there are other important considerations. First is margin control. This will enable you to change margins, alter page length and produce some special effects. Next comes control over the format of the lines so that spacing between lines can be manipulated, and justification and filling can be turned on and off. Carriage con-

Low Cost		Better System	
Mits 88-1302	3495	8800B/T	1295
2 minifloppys	-0-	2 Floppy	2695
2 16K Dynamic	-0-	3 16K Static	2355
1 16K Dynamic	395		
ADM 3	895	B100/lowercase	1595
CENTRONIX	2375	Qume	2995
4.1 BASIC	200	4.1 BASIC	200
	$7360		$11,135

Example 8. System-configuration comparisons.

DOCUFORM program listing.

```
100 '        ============ WRITTEN BY D. L. FITCHHORN ============
110 '      =  ****   ***   ***  *   * *****  ***  ****  *   *  =
120 '     =   *    * *   * *   * *   * *      *   * *   * ** **   =
130 '    =    *    * *   * *     *   * ****  *    * ****  * * *     =
140 '     =   *    * *   * *   * *   * *      *   * *   * *   *   =
150 '      =  ****   ***   ***   ***  *       ***  *   * *   *  =
160 '        ==== COMPUTING SERVICES PROGRAMMER  - MITS/PCC ====
170 '
180 '                      FILES
190 '    1 - INDEX                        2 - OUTPUT
200 '    3 - DATA                         4 - DICTNARY
210 '    5 - MAIN INPUT                   6-15 - STANDARD PARAGRAPHS
220 '
**********************************************************************
      INITIALIZE

230 PRINT"D O C U F O R M   Version  3.0":
    PRINT"(c) 1978 by: Don Fitchhorn MITS":PRINT
240 CLEAR7000:CLEAR FRE(0)
250 DEFINT A-Z:DIM CO$(150),N2$(15),K(50),LS(100),RS(100)
260 N1$="":LINEINPUT"FILE NAME ?";N1$:IF N1$="" THEN CLEAR200:END
270 Q=5:GOSUB7310
280 A$="":
    INPUT"CRT, PRINTER, OR FILE - TYPE C, P, OR F";A$:
    IF A$="" THEN 260
290 IF A$="P" THEN OP=1:PRINT"PRINTER":GOTO370
300 IF A$<>"F" THEN OP=3:PRINT"CRT":GOTO370
310 OP=2:N1$="":LINEINPUT"FILENAME?";N1$:IF N1$="" THEN 280
320 Q=2
330 ON ERROR GOTO350:GOSUB 7310
340 PRINT"ERROR: ";N1$;" ALREADY EXISTS ON DISK";N1:GOSUB7370:
    LINEINPUT"DO YOU WANT TO OVERWRITE IT ? ";A$:
    IF LEFT$(A$,1)="Y" THEN 360 ELSE 310
350 IF ERR=53 THENRESUME360
360 ON ERROR GOTO0:CLOSE2:OPEN"O",#2,N1$,N1:PRINT#2,""
370 L1=70:T1=70:L2=0:T2=0:L3=L1-L2:H1=1:P=66:
    IN=0:IM=0:CO1=0:DF=0:L9=1:B1=0:BM=6:TM=0:
    F=5:NJ=1:PP1=10:II$=CHR$(9)
380 A9$=".^^.IT.PL.RP.DF.MA.WT.LM.RM.HD.CE.NF.FL.BM.TM.JU"+
        ".NJ.SP.LF.TP.ST.PK.WA.BL.PA.CO.IN.NI.NX.CH.SY.RS.LS"+
        ".SM.EM.NM.SH.OM.CM.OV"
390 A$="":B$="":C$="":H$=""
```

```
980 '
**********************************************************************
        PROCESS A LINE

990 '
------------------------
        GET NEXT LINE

1000 IF EOF(F)
        THEN IF OU
                THEN F=F-1:RETURN
                ELSE GOSUB5620:
                     IF F<6 THEN 7000
                        ELSE CLOSE F:N2$(F)="":F=F-1:GOTO1000
1010 LINEINPUT#F,A$:IFA$="" THEN1000
1020 Z=INSTR(A$,"'"):IF Z THEN Z$=LEFT$(A$,Z-1):A$=MID$(A$,Z+1)
                          ELSE PRINT"NO ' IN ";A$:Z$=LEFT$(A$,5)
DOCUFORM LISTED April 25, 1978

1030 '
--------------------------------
        CHANGE TABS TO BLANKS

1040 II=INSTR(A$,II$):
     IF II
        THEN MID$(A$,II,1)=" ":
             A$=LEFT$(A$,II)+SPACE$(7-(II-1)MOD8)+MID$(A$,II+1):
             GOTO1040
1050 '
--------------------------------
        ADD WORDS INTO INDEX

1060 IF IN=0 THEN 1120
1070 I2=INSTR(A$,"^"):IF I2=0 THEN 1120
1080 I3=INSTR(I2,A$," "):IF I3=0 THEN I3=LEN(A$)
1090 PRINT#2,MID$(A$,I2+1,I3-I2)+CHR$(127)+STR$(H1):
     IF I2>1
        THEN A$=LEFT$(A$,I2-1)+MID$(A$,I2+1)
        ELSE A$=MID$(A$,2)
1100 IM=IM+1:GOTO1070
1110 '
--------------------------------------
        INPUT FROM DATA FILE OR CRT

1120 IF DF=0 THEN 1190
1130 D=INSTR(A$,"??"):IF D=0 THEN 1190
1140 IF N2$(3)="" THEN LINEINPUT"?";G$:GOTO1170
1150 IF EOF(3)
        THEN IF F=5
                THEN PRINT"OUT OF DATA":CLOSE:END
                ELSE CLOSE3:DF=0:CLOSEF:F=F-1:GOTO1000
1160 LINEINPUT#3,G$:G$=MID$(G$,INSTR(G$,"'")+1)
1170 IF D=1
        THEN A$=G$+MID$(A$,D+2):GOTO1120
        ELSE A$=LEFT$(A$,D-1)+G$+MID$(A$,D+2):GOTO1120
1180 '
---------------------------
        CHECK FOR COMMAND

1190 IF LEFT$(A$,1)="." THEN 5000
1200 '
----------- PRINT A$ IF NOT FILLING

1210 IF FL THEN C$=A$:GOSUB 6000 :GOTO1000
1220 '
----------- ADD A$ TO B$

1230 IF LEN(B$)+LEN(A$)>255
        THENKQ=240-LEN(B$):B$=B$+LEFT$(A$,KQ):A$=MID$(A$,KQ+1):GOTO1270
1240 IF RIGHT$(B$,1)<>" "THEN B$=B$+" "
1250 B$=B$+A$:A$=""
1260 '
--------------------------
        FILL C$ FROM B$

1270 IFLEFT$(B$,1)=" "THENB$=MID$(B$,2):GOTO1270
1280 IF LEN(B$)<L3
        THEN IF LEN(A$) THEN 1250 ELSE 1000
1290 GOSUB2240:IF L3 = LEN(C$) THEN GOSUB9000:NJ=0:GOSUB6000:GOSUB9010:GOTO1280
1990 '
---------------------
        JUSTIFY C$

2000 K=1:K1=1
2010 IF OP>1
        THEN IF NJ>1 THEN NJ=1
2020 IF L4<0 THEN L3=L3+L4:L$=LEFT$(C$,ABS(L4)):C$=MID$(C$,ABS(L4)+1)
2030 K4=L3-LEN(C$)
2040 ON NJ GOSUB2080,2170,2210
2050 IF L4<0 THEN L3=L3-L4:C$=L$+C$
2060 GOSUB6000:GOTO1280
2070 '--------------- SUB: ADD BLANKS FROM LEFT AND RIGHT
2080 K=-1
2090 IF K1 THEN K=K+1:K1=INSTR(K1+1,C$," "):K(K)=K1:GOTO2090
2100 KE=K4\K:KF=K4-KE*K
2110 IF K7=0 OR K=1 OR K=K4 OR KF<1
        THEN K7=1:K0=1:KF=K-KF
        ELSE K7=0:K0=-1:KE=KE+1
2120 E$="":K1=1
2130 FOR I=0 TO K-1:
```

trol on the output is another important capability. Finally, there are the special features provided by the DOCUFORM system.

Table 1 is a list of the DOCUFORM commands. Each command has two letters preceded by a period. As shown in the format column of the table, most of the commands are followed by numbers or data when they are used. Example 7 shows how commands are embedded in the text file to control the printing of that file. Note that although there can be more than one command in a line, commands cannot follow text on the same line.

Margin control. DOCUFORM's margin control commands control the top, bottom, left and right margin settings; page length; margin adjustments; and left side of the page. TOP MARGIN (.TMn), BOTTOM MARGIN (.BMn), LEFT MARGIN (.LMn) and RIGHT MARGIN (.RMn) each set the value for one margin. PAGE LENGTH (.PLn) sets or changes the number of lines per page. MARGIN ADJUST (.MAm) moves both the left and right margins in (n) or out (-n) characters.

Also included in margin control are RIGHT SIDE (.RSn {,n,n,n . . .}) and LEFT SIDE (.LSn{,n,n,n . . .}), which set up arrays of right and left margins to allow "moving margins" (see Example 7). NEW MARGIN (.NMn,n) saves the old margins and sets up new ones. This is used primarily for STANDARD HEADERS. OLD MARGIN (.OM) resets the margins saved by a NEW MARGIN.

The following default values are assumed if none are specified:

LEFT MARGIN	=	0
RIGHT MARGIN	=	70
TOP MARGIN	=	0
BOTTOM MARGIN	=	6
PAGE LENGTH	=	66

With these settings and standard paper (85 columns by 66 lines), documents will be printed with zero spaces to the left of the text, 15 spaces to the right of the text, starting at the top of the page, and six blank

```
        E$=E$+MID$(C$,K1,K(I)-K1)+SPACE$(KE):
        KF=KF-1:K1=K(I):
        IF KF=0 THEN KE=KE+KO
2140 NEXT:C$=E$+MID$(C$,K(K-1))
2150 RETURN
2160 '------------------ SUB: SPREAD BLANKS
2170 K5=LEN(C$):K=0
2180 K1=INSTR(K1+1,C$," "):
     IF K1 THEN K=K+1:K(K)=K1:GOTO2180
2190 GOSUB2220:KF=K(KF):RETURN
2200 '---------------- SUB: SPREAD CHARACTERS
2210 K=LEN(C$):K5=K
2220 KD=PEEK(15500):KG=KD*(L3-K5):KE=(KG\K)+1:KF=KG MOD K:RETURN
2230 '
------------------- SUB: FILL C$

2240 IF L3>=LEN(B$) THEN C$=B$:B$="":RETURN
2250 FOR I=L3+1 TO 2 STEP -1
2260   IF MID$(B$,I,1)=" "
          THEN IF MID$(B$,I-1,1)<>" " THEN2300
2270 IF MID$(B$,I,1)="-"
          THEN IF I<=L3
                  THEN C$=LEFT$(B$,I):B$=MID$(B$,I+1):RETURN
2280 NEXT
2290 GOSUB7370:PRINT"ERROR: LINE TOO LONG FOR MARGINS. ";
     L3;"SPACES BETWEEN MARGINS.":PRINT"LINE TO BE SPLIT IS:":
     PRINTZ$:PRINTB$:STOP
2300 C$=LEFT$(B$,I-1):B$=MID$(B$,I+1)
2310 IF MID$(B$,1,1)=" " THEN B$=MID$(B$,2):GOTO2310
2320 C=LEN(C$):
     IF L3-C<6 OR SY=0 THEN RETURN
2330 '
---------------- PERFORM SYLLABICATION

2340 D=INSTR(B$," "):
     IF D THEN D$=LEFT$(B$,D-1)
          ELSE D$=B$
2350 IF RIGHT$(D$,1)="." THEN RETURN
2360 D=LEN(D$):
     IF D<9 THEN RETURN
2370 D1$=""
2380 P1=0:
     FOR I=1 TO D:
       A=ASC(MID$(D$,I)):A=A+32*(95<A)
2390   IFA<65 OR 95<A THEN 2410
2400   D1$=D1$+CHR$(A):P1=P1+A
2410 NEXT:
     P1=(P1 MOD 101)+2
2420 D=LEN(D1$):
     IF D<9 THEN RETURN
2430 GET4,P1:E=INSTR(R$,D1$)
2440 IF E=0
        THEN Q=CVI(RR$):
               IF Q THEN P1=Q:GOTO2430
                  ELSE 2480
2450 F1=E+D+1:E$=MID$(R$,F1,INSTR(F1,R$,CHR$(125))-F1)
2460 FOR I=LEN(E$) TO 1 STEP-1:
       G=ASC(MID$(E$,I)):
       IF C+G<L3
          THEN C$=C$+" "+LEFT$(B$,G-1)+"-":B$=MID$(B$,G):RETURN
          ELSE NEXT
2470 RETURN
2480 GOSUB7370:
     PRINT:D2$="--> "+D1$+" <--":D2$=SPACE$((48-LEN(D1$))\2)+D2$:
     PRINTD2$:PRINT"        IS NOT IN MY HYPHENATION DICTIONARY"
2490 PRINT"PLEASE TYPE IT IN WITH DASHES BETWEEN SYLLABLES."
2500 G$="":LINEINPUT"?";G$:IF G$="" THEN RETURN
2510 E$="":J=0:G=1
2520 H=INSTR(G,G$,"-"):
     IF H THEN E$=E$+CHR$(H-J):J=J+1:G=H+1:GOTO2520
2530 F$=D1$+CHR$(124)+E$+CHR$(125)+CHR$(126)
2540 G$=LEFT$(R$,INSTR(R$,CHR$(126))-1)
2550 IF LEN(G$)+LEN(F$)<125
        THEN LSETR$=G$+F$:PUT#4,P1:
               GET4,1:AB=AB+1:LSETAB$=MKI$(AB):PUT4,1:
               GOTO2460
2560 LSETRR$=MKI$(AA):LSETR$=F$:PUT#4,P1:
     AA=AA+1:LSETR$=" ":LSETRR$=MKI$(AA):PUT#4,1
2570 GOTO2460
2580 '
**********************************************************************
        SUB: MAKE CHARACTER SUBSTITUTIONS

2590 IF CH=0 THEN RETURN
2600 CI=0:
     FOR CG=1 TO CH
2610   CI=INSTR(CI+1,C$,MID$(CH$,CG,1))
2620   IF CI THEN MID$(C$,CI)=MID$(CI$,CG,1):GOTO2610
2630 NEXT:RETURN
2640 '
**********************************************************************
        SUB: HOF & HEADER

2650 IF L9=TM+1
        THEN 2670
        ELSE FOR L9=L9 TO P+TM:
                GOSUB6110:
             NEXT:H1=H1+1
2660 IF WT THEN LINE INPUT"WAIT -- ALIGN PAPER";ZZ$
2670 Z9=B1:GOSUB9000:L9=TM+1:IF H$="NO" THEN2720
```

lines at the bottom. By changing the position of the paper in the printer these spacings can be changed to: seven spaces to the left, eight spaces to the right, one blank line at the top and five blank lines at the bottom. Since it is often difficult (impossible) to laterally align the paper to achieve a desired number of spaces to the left, DOCUFORM provides the OVER (.OVn) command, which tells DOCUFORM to move the text over n spaces when it prints.

Formatting. The formatting commands can be divided into three basic types.

1. Commands that expect numbers.
2. Commands that expect data.
3. Commands that act as switches.

The commands that expect numbers are: BLANK LINE (.BL{n}) (number of lines to skip); INDENT (.ITn) (number of space to indent—may be negative); JUSTIFY (.JUn) (type of justification—1, 2 or 3); SPACING (.SPn) (1-single, 2-double, etc.); and SYLLABICATION (.SYn) (0-off, 1-on). The commands that expect data are: CENTER LINE (CEd) (the line to be centered); and LINE FORMAT (.LFd) (the line to be formatted). The commands that act as switches are: FILL (.FL) (turns on fill mode); NO FILL (.NF) (turns off fill mode); and NO JUSTIFY (.NJ) (turns off justify).

Carriage Control. The carriage control commands can also be divided into the three basic types. The commands that expect numbers are: TEST PAGE (.TPn) (tests for n lines left on page—if n lines are not available it performs a page command); PAGE (.PA{n}) (forces page eject—changes page number to n if n is specified); and WAIT AT TOP (.WTn) (waits at top of each page for paper change—0-on, 1-off).

The HEADER (.HDd) command expects data. The characters following the header command will be printed at the top of each page. If the header is empty, the page number is

printed alone. Setting header to the two letters "NO" turns off page numbering. WAIT (.WA) simply waits for the operator to type return.

Special features. The special feature commands are divided into five different types. The three basic types and: 4—commands that expect a file name, and 3—special characters.

The POKE (.PKn,n) command takes two numbers: the first is the location to be POKEd, and the second is the POKE value. Thus .PK15515,205 will have the same effect as an Altair BASIC POKE 15515,205 command. This command can be used to get special printer effects, such as changing the number of characters printed per inch (on a Q70 printer).

CHANGE CHARACTER (.CHcn) takes a character (what you want to change) and a number (the ASCII value to which it will be changed). DOCUFORM performs the specified translations just before printing each line.

Sometimes, instead of having to specify the same group of commands over and over, it is convenient to be able to specify the group with a single command. The SET MACRO (.SMo.a{.a.a.a}) command takes a number (macro number) and a list of commands. EXECUTE MACRO (.EMo) takes just a number (macro number). .EM7 will perform all of the commands set up by a .SM7. xx.xx.xx.

The CONTENTS (.COd) command expects data. This data, along with the current page number, will be saved until the table of contents is printed at the end. The COMMENT (.CM {?}d) command is text that is ignored unless the first character is a "?", in which case the comment will be printed on the terminal.

INDEX (.IN) and NO INDEX (.NI) turn the index checking routine on and off. REPEAT FILE (.RP) closes the current file and then opens it again at the beginning. This is handy for form letters.

DATA FILE (.DF{f}), NEXT FILE (.NFf), STANDARD HEADER (.SHof) and STANDARD

```
2680 IF H$=""
        THEN C$=SPACE$(L3-10)
        ELSE IF L3-10<LEN(H$)
                THEN C$=""
                ELSE C$=SPACE$(L3-(LEN(H$)+10))
2690 C$=C$+H$+"page"+STR$(H1)
2700 IF B1<1 THEN B1=1
2710 H4=L4: NJ=0: L4=0: GOSUB6010: L4=H4
2720 GG$=A$: X1=FL: X2=SY: X3=LM: LM=0: X4=T1: X5=T2: X6=KD: X7=KE: X8=KF
2730 OU=1: FORGG=0TO9: IFLEN(SH$(GG))THENA$=SH$(GG): GOSUB5420
2740 NEXT: OU=0
2750 A$=GG$: FL=X1: SY=X2: LM=X3: T1=X4: T2=X5: KD=X6: KE=X7: KF=X8: B1=Z9: GOSUB9010
2760 RETURN
4990 '
**********************************************************************
        SUB:  COMMAND HANDLER

5000 K7=0
5010 GOSUB5620: AC=ASC(MID$(A$,2)): IF AC>96THEN MID$(A$,2,1)=CHR$(AC-32):
                MID$(A$,3,1)=CHR$(ASC(MID$(A$,3))-32)
5015 AC=ASC(MID$(A$,2)): IF AC>96THEN MID$(A$,2,1)=CHR$(AC-32):
                MID$(A$,3,1)=CHR$(ASC(MID$(A$,3))-32)
5020 A9=INSTR(A9$,MID$(A$,1,3))\3
5030 ON A9 GOTO 5110,5140,5150,5170,5190,5210,5220,5230,5240,5260,5270,
                5280,5310,5320,5330,5340,5380,5390,5400,5420,5430,5440,
                5450,5470,5490,5500,5510,5520,5530,5560,5080,5050,
                5290,5300,5350,5360,5370,5120,5250
5040 PRINTMID$(A$,1,3)," IS A BAD COMMAND ON LINE "; Z$; : GOSUB7370: GOTO5590
5050 LM=0: L8=0
5060 L8=INSTR(L8+1,A$,","): IF L8 THEN LS(LM)=VAL(MID$(A$,L8+1)): LM=LM+1: GOTO5060
5070 LS(LM)=0: L8=0: GOTO5220
5080 RM=0: R8=0
5090 R8=INSTR(R8+1,A$,","): IF R8 THEN RS(RM)=VAL(MID$(A$,R8+1)): RM=RM+1: GOTO5090
5100 RS(RM)=0: R8=0: GOTO5230
5110 L4=VAL(MID$(A$,4)): L3=L3-L4: GOTO5590
5120 IFMID$(A$,4,1)="?"THENPRINTMID$(A$,5)
5130 GOTO5590
5140 P=VAL(MID$(A$,4)): GOTO5590
5150 CLOSE F: Q=F: N1$=N2$(F)
5160 GOSUB7310: GOTO5590
5170 N1$=MID$(A$,4): DF=1:
     IF (LEN(N1$))AND(N1$<>N2$(3))
        THEN Q=3: GOSUB7310
5180 N2$(3)=N1$: GOTO5590
5190 T3=VAL(MID$(A$,4)):
     IF T3
        THEN L1=L1-T3: L2=L2+T3
        ELSE L1=T1: L2=T2
5200 L3=L1-L2: GOTO5590
5210 WT=VAL(MID$(A$,4)): GOTO5590
5220 L2=VAL(MID$(A$,4)): T2=L2: L3=L1-L2: GOTO5590
5230 L1=VAL(MID$(A$,4)): T1=L1: L3=L1-L2: GOTO5590
5240 H$=MID$(A$,4): GOTO5600
5250 OV=VAL(MID$(A$,4)): GOTO5590
5260 C$=SPACE$(INT((L3-LEN(MID$(A$,4)))/2))+MID$(A$,4):
     GOSUB9000: NJ=0: GOSUB6000: GOSUB9010: GOTO5600
5270 FL=1: GOTO5590
5280 FL=0: GOTO5590
5290 I=VAL(MID$(A$,4)): MA$(I)=MID$(A$,5): GOTO5600
5300 I=VAL(MID$(A$,4)): A$=LEFT$(A$,4)+MA$(I)+MID$(A$,5): GOTO5590
5310 BM=VAL(MID$(A$,4)): GOTO5590
5320 TM=VAL(MID$(A$,4)): GOTO5590
5330 NJ=VAL(MID$(A$,4)): GOTO 5590
5340 NJ=0: GOTO 5590
5350 O3=L3: O4=L4: O2=L2: O1=L1: L2=VAL(MID$(A$,4)):
     L1=VAL(MID$(A$,INSTR(A$,",")+1)): L3=L1-L2: L4=0: GOTO5590
5360 I=VAL(MID$(A$,4)): SH$(I)=".ST"+MID$(A$,5): GOTO5590
5370 L3=O3: L4=O4: L2=O2: L1=O1: GOTO5590
5380 B1=VAL(MID$(A$,4))-1: GOTO5590
5390 C$=MID$(A$,4): K=1: K1=1: K4=L3-LEN(C$): GOSUB9000: NJ=0:
     GOSUB2080: GOSUB6000: GOSUB9010: GOTO1000
5400 IF L9>P-BM-VAL(MID$(A$,4))
        THEN GOSUB2650
5410 GOTO5590
5420 F=F+1: N1$=MID$(A$,4): Q=F: GOSUB7310: GOTO5590
5430 POKE VAL(MID$(A$,4)),VAL(MID$(A$,INSTR(A$,",")+1)): GOTO5590
5440 PRINT"WAIT "; MID$(A$,4); : LINEINPUT" "; ZZ$: GOTO5590
5450 B9=B1: B1=VAL(MID$(A$,4)): IF B1=0 THEN B1=B9: GOTO5590
5460 GOSUB6040: B1=B9: GOTO5590
5470 P9=VAL(MID$(A$,4)): IF P9 THENH1=P9-1
5480 GOSUB2650: GOTO5590
5490 CO$(CO1)=MID$(A$,4)+CHR$(127)+STR$(H1): CO1=CO1+1: GOTO5590
5500 IN=1: OPEN"O",1,"INDEX": GOTO5590
5510 IN=0: GOTO5590
5520 N1$=MID$(A$,4): Q=F: GOSUB7310: GOTO5600
5530 CG$=MID$(A$,4,1): CG=VAL(MID$(A$,5)):
     IF ASC(CG$)<>CG
        THEN CH$=CH$+CG$: CI$=CI$+CHR$(CG): CH=LEN(CH$): GOTO5590
5540 CF=INSTR(CH$,CG$):
     IF CF=1
        THEN CH$=MID$(CH$,2): CI$=MID$(CI$,2)
        ELSE IF CF=CH AND CH<>0
                THEN CH$=LEFT$(CH$,CF-1): CI$=LEFT$(CI$,CF-1)
                ELSE CH$=LEFT$(CH$,CF-1)+MID$(CH$,CF+1):
                     CI$=LEFT$(CI$,CF-1)+MID$(CI$,CF+1)
5550 CH=LEN(CH$): GOTO5590
5560 SY=VAL(MID$(A$,4)): CLOSE4: IF SY=0THEN5590
5570 OPEN"R",4,"DICTNARY": FIELD#4,126 AS R$,2 AS RR$:
FIELD#4,2AS AA$,2AS AB$,116ASG$,8ASG$: GET#4,1
5580 IF G$="DICTNARY"THEN AA=CVI(AA$): AB=CVI(AB$): GOTO5590
```

```
ELSE LSETG$="DICTNARY":LSETAA$=MKI$(102):LSETAB$=MKI$(0):PUT4,1:
LSETR$=CHR$(125)+CHR$(126):LSETRR$=MKI$(0):FORM=2TO101:PUT4,M:NEXT:GOTO5590
5590 A9=INSTR(2,A$,"."):
     IF A9<>0
        THEN A$=MID$(A$,A9):GOTO5015
5600 B$="":GOTO1000
5610 '
---------------- SUB: RUN OUT B$

5620 IFB$=""THENRETURN
5630 C$=B$:B$="":GOSUB9000:NJ=0
5640 IF (NF=0)AND(L9+B1<P)
        THEN GOSUB6010 ELSE GOSUB6000
5650 GOSUB9010:RETURN
5990 '
***********************************************************************
        SUB: PRINT A LINE

6000 IF L9>P-BM THEN GH$=B$:GK$=C$:GOSUB2650:B$=GH$:C$=GK$
6010 GOSUB2590
6020 GOSUB6170:IF LM=0 THEN L3=L3+L4
6030 L4=0:L9=L9+1
6040 IF B1=0 THEN GOTO6090
6050 FOR B2=1 TO B1
6060    GOSUB6110
6070    L9=L9+1
6080 NEXT
6090 C$="":RETURN
6100 '
***********************************************************************
        SUB: OUTPUT BLANK LINE TO CORRECT DEVICE

6110 ON OP GOSUB 6130,6140,6150
6120 GOTO6180
6130 LPRINT:RETURN
6140 PRINT#2,PP1;"'":PP1=PP1+10:RETURN
6150 PRINT:RETURN
6160 '
***********************************************************************
        SUB: SEND OUTPUT TO CORRECT DEVICE

6170 ON OP GOSUB 6200,6300,6310
6180 IF LM
        THEN IF LS(L8)=0 AND RS(L8)=0
                THEN LM=0
                ELSE L2=LS(L8):L1=RS(L8):L3=L1-L2:L8=L8+1
6190 RETURN
6200 LPRINTTAB(L2+L4+OV):
     IF L4>0
        THEN L5=1
        ELSE L5=ABS(L4)+1
6210 LPRINT LEFT$(C$,L5);
6220 IFFL=OTHENON NJ GOTO 6230,6270,6240
6230 LPRINTMID$(C$,L5+1):RETURN
6240 POKE15500,KD+KE:POKE15517,KD+KE:LPRINTMID$(C$,L5+1,KF);
6250 POKE15500,KD+KE-1:POKE15517,KD+KE-1:LPRINTMID$(C$,KF+L5+1)
6260 POKE15500,KD:POKE15517,KD:RETURN
6270 IF KF=0 THEN KF=1
6280 POKE15500,KD+KE:LPRINTMID$(C$,L5+1,KF);
6290 POKE15500,KD+KE-1:LPRINTMID$(C$,KF+L5+1):POKE15500,KD:RETURN
6300 PRINT#2,PP1;"'";SPACE$(L2+L4+OV)+C$:PP1=PP1+1:RETURN
6310 PRINTTAB(L2+L4+OV);C$:RETURN
6980 '
***********************************************************************
        FINAL OUTPUT

6990 '
------------------- CLOSE FILES

7000 IF IN THEN PRINT#2,"END"
7010 CLOSE
7020 '
------------------- SORT AND PRINT INDEX

7030 IF IM=0 THEN GOTO 7220
7040 IM=IM-1
7050 DIMIN$(IM):CLOSE:OPEN"I",1,"INDEX"
7060 FORI=OTOIM:LINEINPUT#1,IN$(I):NEXT
7070 FOR H=IM-1 TO 0 STEP-1:N=H:
       FOR K=0 TO H:
         IF IN$(K)<IN$(N)
           THEN NEXT
           ELSE N=K:NEXT
7080   SWAPIN$(H),IN$(N):
     NEXT
7090 C$=SPACE$(INT((L3-5)/2))+"INDEX":GOSUB6000
7100 B1=0
7110 J=1
7120 IF J>IM THEN 7170
7130 J1=INSTR(IN$(J-1),CHR$(127)):J2=INSTR(IN$(J),CHR$(127))
7140 IF LEFT$(IN$(J-1),J1)<>LEFT$(IN$(J),J2)
       THEN 7160
       ELSE IF IN$(J-1)<>IN$(J)
              THEN IN$(J-1)=IN$(J-1)+MID$(IN$(J),J2)
7150 FORK1=JTOIM-1:
       IN$(K1)=IN$(K1+1):
     NEXT:IN$(IM)="":IM=IM-1
7160 J=J+1:GOTO7120
7170 FOR J=0TOIM-1
7180   C$=IN$(J)
```

PARAGRAPH (.STf) all expect a file name, although DATA FILE can be specified with no file name to request data from the CRT. DATA FILE can be used along with REPEAT FILE to personalize form letters. The data in a data file may contain commands that will be executed if the ?? is in a command line.

NEXT FILE causes DOCUFORM to close the file it is working on, open the file specified and begin processing it. This allows an index across several files. Or, if you run out of room on one disk, the rest of the file can be kept on another disk.

The STANDARD PARAGRAPH and STANDARD HEADER commands act like NEXT FILE except that when the called file is finished, the calling file continues from where it left off. A standard paragraph can call other standard paragraphs (which can call other standard paragraphs . . .).

If you want to produce legal forms and other documents that have the same old things stated over and over, STANDARD PARAGRAPH will save you a lot of typing. Up to ten (0-9) STANDARD HEADERS may be active at one time. Each header will execute every time the program executes the HEAD OF FORM routine. To turn on STANDARD HEADER #3, execute (.SH3filename). To turn it off, execute (.SH3).

There are three special characters in DOCUFORM: period, double question mark and up-arrow. Period, if it is the first character on a line, tells the program that the line is a command line. It also separates multiple commands on one line.

Double question mark (??) tells DOCUFORM to pick up data and can appear anywhere in a line. Each ?? will be replaced by a line from the data file or terminal. (Make sure you use a .DF{f} command before using ?? or you will just get ?? printed.) Up-arrow (Λ) indicates, if a .IN command has been executed, that the following word is to be put into the index. The up-arrow is deleted by DOCUFORM and will not show in the index or the printed document.

These commands may be used in a variety of ways to produce interesting effects. See Example 7 for a sample text file using many of these commands. It takes one input file and prints it four different ways.

System Configuration

I am running DOCUFORM on an Altair 8800B with 48K of memory, four Altair disks, 4.1 Disk BASIC, B100 terminal and Q70 line printer. Three of the disks are not necessary for this program, and the B100 terminal could be replaced by an ADM-3. This would reduce the system cost considerably. The Q70 printer is a necessity if you want type 2 or 3 justification. Using this system configuration and justification mode 3, I can run the Q70 at almost full speed—45 characters per second or about 270 wpm.

The configurations in Exam-

```
7190   GOSUB2590: GOSUB6000: C$=""
7200 NEXT
7210 '
-------------------- PRINT TABLE OF CONTENTS

7220 IFCO1<1THENGOTO7290
7230 H$="": GOSUB2650: C$="TABLE OF CONTENTS": GOSUB6010
7240 FORJ=0TOCO1-1
7250 I=INSTR(CO$(J), CHR$(127))
7260 C$=LEFT$(CO$(J), I-1)+STRING$(L3-LEN(CO$(J)), 46)+MID$(CO$(J), I+1)
7270 GOSUB6010
7280 NEXT
7290 GOSUB2650: RUN
7300 '
************************************************************************
       SUB: OPEN FILES

7310 CLOSE Q: N2$(Q)=N1$
7320 IF RIGHT$(N1$, 1)=" "
       THEN N1$=LEFT$(N1$, LEN(N1$)-1): GOTO7320
7330 IF LEFT$(RIGHT$(N1$, 2), 1)=","
       THEN N1=VAL(RIGHT$(N1$, 1)): N1$=LEFT$(N1$, LEN(N1$)-2)
       ELSE N1=0
7340 OPEN"I", Q, N1$, N1: LINEINPUT#Q, G$
7350 IF 0<LEN(G$)
       THEN PRINTN1$" ON DRIVE"; N1; " IS NOT AN ASCII SAVED FILE";:
            GOSUB7370: STOP
       ELSE RETURN
7360 '
************************************************************************
       SUB: BEEP AN ERROR

7370 FOR J=1 TO 3:
      PRINT"! ";:
      FOR I=1 TO 5:
        PRINTCHR$(7);:
      NEXT:
      FOR I=1 TO 900: NEXT:
    NEXT: RETURN
9000 U1=U1+1: NJ(U1)=NJ: RETURN
9010 NJ=NJ(U1): U1=U1-1: RETURN
```

Variable	Description
A	- ASCII Value of Character in Word to be Hyphenated
A$	- Input Buffer
A9	- Command Indicator
A9$	- Command List
AA	- Pointer to Next Overflow Record in Dictnary
AA$	- Area in field statement for Next Overflow Record Pointer in DICTNARY
AB	- Number of Words in DICTNARY
AB$	- Area in Field Statement for number of words in DICTNARY
B$	- Data Buffer
B1	- Spacing
B2	- Loop Counter
B9	- Temporary Storage for B1
BM	- Bottom Margin
C	- Length of C$ During Syllibication
C$	- Output Buffer
CF	- Location of Character to be Pulled out of CH$
CG	- Loop Counter and Test Variable
CG$	- Temp Change Character Storage
CH	- # of Characters in CI$ and CH$
CH$	- Change from String
CI	- Temp Location Variable
CI$	- Change to String
CO$()	- (150) Contents Array
CO1	- Pointer into CO$
D	- Temporary Variable
D$	- Word Being Hyphenated
D1$	- Temporary String
D2$	- Temporary String
DF	- Data Input Flag
E	- Temporary Variable
E$	- Hyphenation Points for Word (from DICTNARY)
F	- File Pointer
F$	- Output String to Add a Word and Its Hyphenation Points into DICTNARY
F1	- Temporary Variable
FL	- Fill/No Fill Flag
G	- Temporary Variable
G$	- Input String
GG	- Standard Header loop counter
GG$	- Temporary storage for A$
GH$	- Temporary storage for B$
GK$	- Temporary storage for C$
H	- Loop Counter
H$	- Header
H1	- Page Number
H4	- Temporary storage for L4
I	- Loop Counter
I2	- Temporary Variable
I3	- Temporary Variable
II	- Temporary Variable
II$	- Tab character
IM	- Index Counter
IN	- Index Flag
IN$	- Index Sort Array
J	- Temporary Variable
J1	- Temporary Variable
J2	- Temporary Variable
K	- Number of Characters to be Spread in Justification
K()	- (50) Space Locations for Justification
K0	- Forward/Backward Variable
K1	- Temporary Variable
K4	- # of Characters Necessary for Justification of Line
K5	- Length of C$
K7	- Forward/Backward Flag
KD	- Temporary Variable
KE	- # of Spaces (or Increments) to be Added Each Time
KF	- # of Times to use KE Before Adding K0 to It.
KG	- Temporary Variable
KQ	- Position to break A$
L$	- Temporary Storage for Left Side of C$
L1	- Right Margin
L2	- Left Margin
L3	- Line Length
L4	- Indent
L5	- Print Variable for Indent
L8	- Comma position in Left Side string
L9	- Line Count
LM()	- Left margin array
LS()	- (100) Left Side array
M	- Loop counter for DICTNARY creation
MA$()	- (10) Macro array
MB	- Temporary storage for B1
MD	- Temporary storage for KD
ME	- Temporary storage for KE
MF	- Temporary storage for KF
N	- Sort Variable
N1	- Drive Number for File
N1$	- File Name
N2$()	- (15) File Names in Use
NJ	- Justify Flag
NJ()	- (10) Last in - first out (LIFO) stack for NJ
O1	- Temporary storage for L1
O2	- Temporary storage for L2
O3	- Temporary storage for L3
O4	- Temporary storage for L4
OP	- Output Device
OU	- Flag for Standard Header
OV	- Spaces to left side of page
P	- Page Length
P1	- Pointer into DICTNARY
P9	- Change Variable for Page #
PP1	- Line # for File Output
Q	- Temporary Variable
Q1$	- Data File Name ("" if CRT)
R8	- Comma position in Right Side string
R$	- Data Field in DICTNARY File
RM()	- Right margin array
RR$	- Overflow pointer in DICTNARY record
RS()	- (100) Right Side array
SH$()	- (10) Standard Header array
SY	- Syllabication Flag
T1	- Hold Area for L1
T2	- Hold Area for L2
T3	- Margin Adjust Amount
TM	- Top Margin
WT	- Wait Flag
U1	- Stack pointer for NJ storage
X1-X9	- Environment preservation storage
Z	- Temporary Variable
Z$	- Line Number for Line being Processed
Z9	- Temporary Variable
ZZ$	- Dummy Input Variable for Wait

Table 2. Variable descriptions.

```
INITIALIZATION
    OPEN FILES
    BEEP AN ERROR
PROCESS A LINE
    GET NEXT LINE
        HOF & HEADER
            SEND OUTPUT TO CORRECT DEVICE
    CHANGE TABS TO BLANKS
    ADD WORDS TO INDEX
    INPUT FROM DATA FILE OR CRT
    CHECK FOR COMMAND
            COMMAND HANDLER
    FILL C$ FROM B$
        FILL C$
    JUSTIFY C$
        PRINT A LINE
            MAKE CHARACTER SUBSTITUTIONS
FINAL OUTPUT
    CLOSE FILES
    SORT & PRINT INDEX
    PRINT TABLE OF CONTENTS
```

Fig. 1. Major program divisions.

ple 8 will get DOCUFORM working for you at your best cost. In my opinion, the better system is really the minimal system, which you should buy if you intend to make heavy use of a computer.

Consider! Buying the better system with a 5-year amortization, and having a liberal allowance for paper and ribbons, you are looking at $3000 per year for a system that can replace several secretaries (or give your secretaries more time to do other things). If your business is too small to justify a computer on its word-processing capabilities alone, remember that it can also handle your inventory, payroll, receivables and who knows what else.

Note: Although DOCUFORM was written in Mits 4.1 Disk BASIC, it should not require any modifications to run on machines with a similar BASIC. The only exception to this is the use of the Mits Qume line printer driver by justification modes 2 and 3.

The Mits driver allows changing the number of the vertical and horizontal increments by which the Qume moves its printhead and platen by poking values into special locations in the driver. Horizontally, the increment is in 128ths of an inch. Vertically, it is in 48ths of an inch. Altering lines 3190, 6090, 6100, 6110, 6130 and 6140 to agree with your Qume driver will be necessary if you are not using the standard version of Mits 4.1 Disk BASIC.

If you don't have a Qume printer on your system yet (they are a bit expensive), don't worry. You can still use DOCUFORM with justification mode 1. Just don't try to use justification modes 2 or 3 or you will have to reload your BASIC.

Program Description

DOCUFORM is divided into three main sections: INITIALIZATION, PROCESS A LINE and FINAL OUTPUT (see Fig. 1). Each section will be discussed along with any subroutines that it may use. See Table 2 for a description of each program variable.

The INITIALIZATION section sets up string space (memory assigned for string variable use during program execution) to whatever is available, minus 7000 bytes. The input file name is read, and the subroutine OPEN FILES is called.

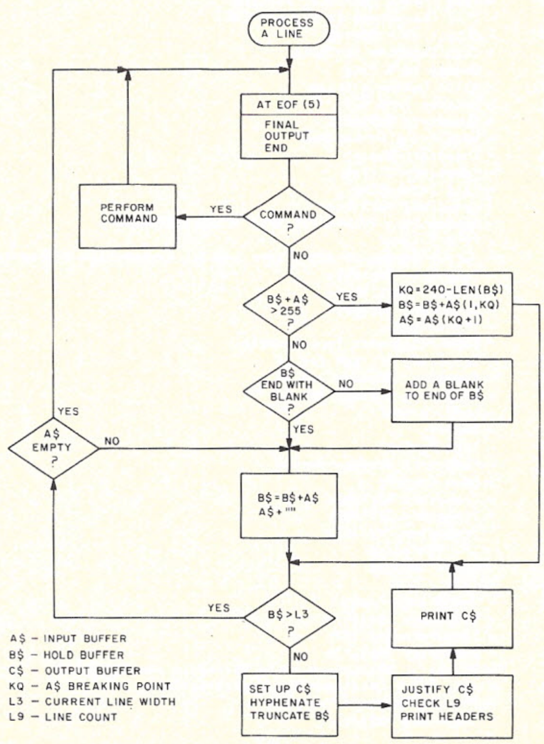
The OPEN FILES subroutine allows storage of the name of all files that are in use at any time, as well as opening these files for input. This will allow DOCUFORM to reopen these files as necessary with no operator interaction. By testing the file name to see if there is a specified disk number at the end, open files can open the file on the correct disk. Before returning, this routine makes sure that the file was saved in ASCII. Files not saved in ASCII cannot be properly printed.

INITIALIZATION now asks where the output is to go. If output to a file is selected, this file is checked to see if it exists. If it does, you are warned by BEEP AN ERROR (which beeps at you) and are asked if you want to overwrite it. Before exit to process a line, INITIALIZATION presets the values of most variables for use later in the program.

The PROCESS A LINE section (see Flowchart 2) is the heart of the program and contains several subsections (as shown in Fig. 1). The GET NEXT LINE subsection must first test for "END OF FILE" on the file being read. If the file is at the end, a check is made to determine if the current file is the main input file. If so, control transfers to the FINAL OUTPUT section. Otherwise, the file is closed, and the file indicator is pointed back to the file that caused this one to be opened. Control is then transferred back to the beginning of the PROCESS A LINE section to work on the next line in that file. If the file is not at the end, the next line is read from it.

The line number and remark character (') are pulled off the front of the line at this point. This enables DOCUFORM to process any ASCII-saved program file. (See AUTO DOC and FILE CORRECTION for an explanation of program files.) The line number is saved so, if an error should occur, it can be printed to show where DOCUFORM was in the input file. The CHANGE TABS TO BLANKS routine simply does what the title says.

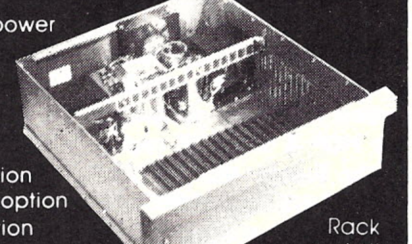
The ADD WORDS INTO INDEX routine is executed if an index has been requested. This routine stores each word preceded by an up-arrow (∧) in the index file along with the cur-



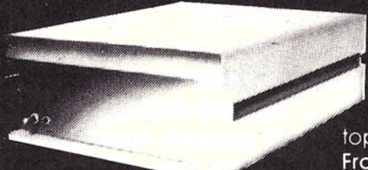
Flowchart 2. Process-a-line section.

S-100 MAINFRAME $200
S-100 MAINFRAME $200

- NOT A KIT
- 8v@15A, ±16v@3A power
- Rack mountable
- 15 slot motherboard
- Card cage
- Fan, line cord, fuse, switch, EMI filter
- Desk top version option
- 8v@30A, ±16v@10A option
- voltage monitor option
- 100 pin connectors optional



Rack mounted model From $200



Desk top model From $235

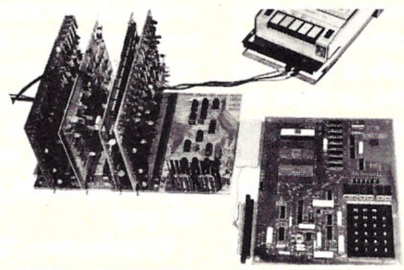
Write or call for a copy of our detailed brochure which includes our application note BUILDING CHEAP COMPUTERS.

INTEGRAND I13

8474 Ave 296 • Visalia, CA 93277 • (209) 733-9288

We accept BankAmericard/Visa and Master Charge

KIMSI



The KIM to S-100 bus Interface/Motherboard

- Combines the power of the 6502 with the flexibility of the S-100 bus
- Attaches to any unmodified KIM
- Complete interface logic and fully buffered motherboard in one unit
- On-board regulation of power for KIM
- Eight slots of S-100 compatibility for additional RAM, Video and I/O boards, PROM Programmers, Speech processors . . .
- Includes all parts, sockets for ICs, one 100 pin connector, and full Assembly/Operating documentation

◆ *Kit $125, Assembled $165*
◆ *All units shipped from stock*

FORETHOUGHT PRODUCTS
P.O. Box 8066-H
Coburg, OR 97401

master charge · VISA · F8

WIREWRAP

PRECUT WIRE

Why buy wire on rolls?

PRECUT & STRIPPED WIRE IS:

- **Fast** - No more cutting & stripping by hand
- **Reliable** - Good, clean, uniform strip
- **Economical** - Cheaper than using bulk wire

Precut Wire	Bulk Wire
100 pcs of 3" at $.82 = **3¼¢/ft.**	50 ft roll at $1.99 = **4¢/ft.**
100 pcs of 6" at 1.06 = **2¢/ft.**	100 ft. roll at 2.95 = **3¢/ft.**
Wire Kit 1 at $6.95 = **2 1/3¢/ft.**	

30 Kynar stripped 1" on each end. Lengths are overall
Colors: Red,Blue,Green,Yellow,Black,Orange,White
Wire packaged in plastic bags. Add 25¢/length for tubes

	100	500	1000	5000
2½ in	.78	2.40	4.30/K	3.89/K
3 in.	.82	2.60	4.71/K	4.22/K
3½ in.	.86	2.80	5.12/K	4.55/K
4 in.	.90	3.00	5.52/K	4.88/K
4½ in.	.94	3.21	5.93/K	5.21/K
5 in.	.98	3.42	6.34/K	5.52/K
5½ in.	1.02	3.65	6.75/K	5.86/K
6 in.	1.06	3.85	7.16/K	6.19/K
6½ in.	1.15	4.05	7.57/K	6.52/K
7 in	1.20	4.25	7.98/K	6.85/K
7½ in.	1.25	4.45	8.39/K	7.18/K
8 in	1.29	4.65	8.80/K	7.53/K
8½ in.	1.32	4.85	9.21/K	7.84/K
9 in.	1.36	5.05	9.62/K	8.17/K
9½ in.	1.40	5.25	10.03/K	8.50/K
10 in	1.45	5.51	10.44/K	8.83/K
Addl. in.	.10	41	82/K	.66/K

WIRE WRAP SOCKETS

	1-9	10-24	25-99	100-249	250-999	1K-5K
8 pin*	.41	.38	.35	.31	.29	.27
14 pin*	.36	.34	.32	.30	.28	.27
16 pin*	.38	.36	.34	.32	.30	.29
18 pin*	63	58	.54	.47	.44	.41
20 pin	84	.78	.71	63	.59	.54
22 pin*	.95	.90	.85	.75	.72	.70
24 pin	91	.84	.78	.68	.64	.59
28 pin	1.25	1.15	1.08	.95	.89	.82
40 pin	1.65	1.55	1.42	1.25	1.15	1.09

Gold 3-level Closed Entry Sockets
End & Side Stackable — All prices include gold
2-level sockets available

WIRE WRAP TOOLS

$34.95

HOBBY WRAP
Model BW 630
With Free Wire Kit 1
($6.95 Value)

Batteries & Charger	$11.00
WSU 30 Hand Wrap-Unwrap Strip Tool	5.95
WSU 30M, for Modified Wrap	6.95
BT 30 Extra Bit	2.95

WIRE KITS

#1 $6.95 — **#2 $19.95**

#1				#2					
250	3"	100	4½"	250	2½"	250	4½"	250	6"
250	3½"	100	5"	500	3"	250	5"	100	6½"
100	4"	100	6"	500	3½"	100	5½"	100	7"
				500	4"		1 250 ft Roll-Bulk		

Choose One Color or Assortment

INTERCONNECT CABLES

Ribbon cable connectors for connecting boards to front panels, or board to board

	SINGLE ENDED			DOUBLE ENDED		
	14 pin	16 pin	24 pin	14 pin	16 pin	24 pin
6"	1.24	1.34	2.05	2.24	2.45	3.37
12"	1.33	1.44	2.24	2.33	2.55	3.92
24"	1.52	1.65	2.63	2.52	2.76	4.31
48"	1.91	2.06	3.40	2.91	3.17	5.08

PAGE DIGITAL ELECTRONICS P36
135 E. Chestnut St. Suite #5
Monrovia, California 91016
Phone (213) 357-5005

Ordering Information:

- Orders under $25 and COD's, add $2
- All others, shipped Ppd in U.S. via UPS
- For Blue Label (Air) or 1st Class, add $1
- We accept Visa & Mastercharge
- Most orders shipped same day

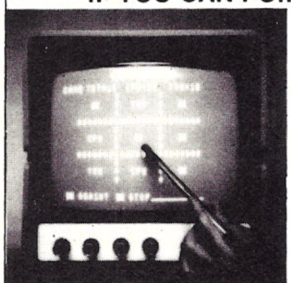
Dealer Inquiries Invited

ELECTRONIC SYSTEMS MARKETING
Presents

VIDIET•STIK™

™

"IF YOU CAN POINT . . . YOU CAN PLAY!"



- ☐ THE NEW LOW-COST ALTERNATIVE TO THE KEYBOARD!
- ☐ FOR ALL Z80 & 8080 BASED MICROPROCESSOR SYSTEMS WITH TVT.
- ☐ REQUIRES ONLY:
  - +5V
  - GROUND
  - SINGLE INPUT BIT

ONLY $39.95*
PLUS $1.50 POSTAGE & HANDLING

APPLICATIONS:

- Keyboard replacement
- Menu selection
- Games (3 yrs. & up)
- Electronic music
- Joystick simulator
- Educational drills

. . . Your imagination's the only limit! . . .

INCLUDES:

- ☐ HARDWARE — Pen is fully assembled and tested!
- ☐ INSTRUCTIONS — Complete interface instructions
- ☐ DOCUMENTATION
  - Driver software
  - Test software
  - Game software

ESMARK INCORPORATED
507½ E. McKINLEY HWY. MISHAWAKA, IN 46544
* INDIANA RESIDENTS ADD 4% SALES TAX — E30

rent page number. The up-arrow is then removed from the line.

Next, if a .DF{f} command has been executed and there are double question marks (??) in the line, input is picked up from a data file (or from the CRT) and added into the string.

If at this point A$ is a command line, control transfers to the COMMAND HANDLER, which, after processing the command(s), goes back to PROCESS A LINE. Otherwise, if A$ is a data line and fill is on (a .FL has been given), A$ is added to the end of the data buffer (B$) and control drops into the FILL C$ FROM B$ subsection.

The FILL C$ FROM B$ subsection strips off leading blanks and then checks the number of characters in the data buffer. If the buffer has less than the maximum line length (L3) characters in it, control is transferred back to GET NEXT LINE to add to the buffer. Otherwise, the FILL C$ routine is called and C$ is set to L3 or less characters from the left side of B$, and B$ gets what is left over.

Syllabication, if requested and necessary, is performed at this time. Note: In order to make the dictionary file as small as possible, all words are converted to uppercase with no non-characters (-/,%.# . . .) embedded.

After C$ is filled with a line to be printed, DOCUFORM prepares to print the line with whatever justification method was specified in the input file.

The line is then printed using the PRINT A LINE routine. The program has data ready to print, so the line count is tested to see if the HOF & HEADER routine should be performed. The HOF & HEADER routine prints (using the OUTPUT BLANK LINE TO CORRECT DEVICE routine) a series of line feeds until the paper is at head of form. Then, if a header was specified, it is printed.

The PRINT A LINE routine next calls the MAKE CHARACTER SUBSTITUTIONS routine, which makes all requested character changes (such as ! to _). Using the SEND OUTPUT TO CORRECT DEVICE, the PRINT A LINE routine finally prints C$ to the desired output device, and control transfers back to FILL C$ FROM B$.

When the entire input file has been processed, control transfers to CLOSE FILES and then to SORT AND PRINT INDEX. If an index was requested, the index file is read in, sorted and printed. A check is then made to determine if a table of contents was requested. If it was, it is printed.

The last statement that DOCUFORM executes is RUN. This gets back to the top with all variables cleared. Back at the top now, DOCUFORM is ready to process another file for you.

AUTO DOC, FILE CORRECTION

AUTO DOC is a program used in conjunction with DOCUFORM. This program allows the entry of text by a typist just as it would normally be typed. Leading blanks are stripped off the entered text, which is then stored on disk with embedded DOCUFORM commands. When the disk file is run through DOCUFORM, it will appear just as the typist entered it. The typist can load and list the file with BASIC to get a feel for how the basic DOCUFORM commands are used.

To aid the typist, AUTO DOC will print, on request, a pair of lines showing the current line number and numbering the columns across the terminal. To request this, the typist should type a CTRL-F. A second CTRL-F will turn this feature back off.

AUTO DOC was written to work on a BEEHIVE 100 video terminal, but can be easily modified to any video terminal by changing the OUTs on line 680 to the back space on your terminal. This line would have to be extensively modified to work on a "hard-copy" terminal, so I would suggest changing it to handle delete in the same manner as the BASIC that you use.

The files created by AUTO DOC are stored in "program file" format (see Example 6) on disk. This enables the file to be loaded using BASIC, and

AUTO DOC program listing.

```
10 PRINT"4/4/78 DON FITCHHORN/GARY RUNYAN ---- AUTO DOC":ONERRORGOTO0
15 CLEAR500:IF FRE(0)<32767 THENCLEAR FRE(0)ELSECLEAR32760
20 ONERRORGOTO10
30 '            ==== WRITTEN BY: D. L. FITCHHORN and GARY RUNYAN ====
40 '          =    *   *   * *****  ***        ****   ***   ***     =
50 '         =    * *  *   *   *   *   *       *   * *   * * · *     =
60 '        =    ***** *   *   *   *   *       *   * *   * *          =
70 '         =   *   * *   *   *   *   *       *   * *   * *   *     =
80 '          =  *   *  ***    *    ***        ****   ***   ***     =
90 '            ======= COMPUTING SERVICES DEPT.  -  MITS/PCC =======
100 '
110 '
120 '    A        =  LEFT MARGIN FOR A$
130 '    A$       =  CURRENT LINE (LINE BEING PROCESSED)
140 '    B        =  LEFT MARGIN FOR B$
150 '    B$       =  NEXT LINE TO BE PROCESSED
160 '    C$       =  COMMANDS LINE
170 '    E        =  NUMBER OF BLANK LINES TYPED BETWEEN CURRENT AND LAST LINE
180 '    G        =  NUMBER OF SPACES TO BE ADDED TO B$ FOR A CTRL-I
190 '    LM       =  CURRENT LEFT MARGIN
200 '    LT       =  SWITCH: 1 = LINE BEFORE A$ TEXT
                             0 = LINE BEFORE A$ COMMAND
210 '    N        =  LINE NUMBER
220 '    X        =  ASCII FOR CHARACTER INPUT BY GET A LINE ROUTINE
230 '

***********************************************************************
         INITIALIZATION

250 DEFINT A-Z:WIDTH79
260 N1$="":LINEINPUT"FILE NAME ? ";N1$:IF N1$="" THEN520
270 N1=0:IF MID$(N1$,LEN(N1$)-1,1) = ","
      THEN A$=RIGHT$(N1$,1):
            IF "0"<A$ AND A$<="9"
               THEN N1$=LEFT$(N1$,LEN(N1$)-2):N1=VAL(A$)
280 CLOSE2:ON ERROR GOTO300:OPEN"I",#2,N1$,N1
290 PRINT"ERROR: ";N1$;" ALREADY EXISTS ON DISK";N1:GOSUB740:
    LINEINPUT"DO YOU WANT TO OVERWRITE IT OR APPEND TO IT ? TYPE O OR A ? ";A$:
    IF LEFT$(A$,1)="O" THEN310ELSE IF LEFT$(A$,1)="A"THEN 760 ELSE260
300 IF ERR=53 THENRESUME310ELSE10
310 ON ERROR GOTO10:CLOSE:OPEN"O",#1,N1$,N1:PRINT#1,"":PRINT#1,"0 '.NF"
320 N=10:INPUT"FIRST LINE #";N
330 PRINT"OK - LAY IT ON ME !!!":PRINT
340 Z$="1234567890":GOSUB550
350 '

***********************************************************************
         PROCESS A LINE

360 C$="":E=0
370 A$=B$:A=B:IFA$=".END"THEN510
380 GOSUB550:IFA$=""THENE=E+1:GOTO370
390 IF E THENC$=".BL"+MID$(STR$(E),2):LT=0
400 IFLEFT$(A$,1)="."THENC$=C$+A$:A$="":GOTO470
410 IFA=LM AND (LT OR B=LM) THEN470
420 IFLEFT$(B$,1)="." OR B$=""
       THENIF A=LM THEN470 ELSEC$=C$+".LM"+MID$(STR$(A),2):LM=A:GOTO470
430 IFLM<>BTHENC$=C$+".LM"+MID$(STR$(B),2):LM=B
440 IFLM<ATHENC$=C$+".IT"+MID$(STR$(A-LM),2)
450 IFLM>ATHENC$=C$+".IT"+STR$(A-LM)
460 '
```

BASIC's editing commands can be used to change it. The corrected file can then be saved back onto the disk. The format for a program file line is:

LINE# 'data..........

The "'" is Mits BASIC short form for REM. Thus the file is made up entirely of remark statements. This keeps BASIC from modifying the line.

If you are interested in other (perhaps better) ways of correcting your DOCUFORM files, you might try EDIT. This program was written for people experienced in using larger machines' editors (DEC's EDIT-11 or TECO, for example). See page 2 of the October 1977 issue (Vol. 3, No. 5) of *Computer Notes* for an explanation of the editor and a program listing.

Conclusion

I believe that the DOCUFORM program is a step in the right direction for microprocessors. Text processing is an area that could benefit many small businesses. However, cost has been a prohibitive factor. Now for less than $12,000 these businesses can compete with owners of systems costing ten timês as much. I hope that you will try DOCUFORM. I think you'll like it as much as my wife does. (Author's note: The last sentence was added by my wife who typed this article for me—using AUTO DOC and DOCUFORM, of course.)■

```
******************************************************************
      OUTPUT TO FILE

470 IF C$<>"" THENPRINT#1,N;"'";C$:N=N+10:LT=0
480 IF A$<>"" THENPRINT#1,N;"'";A$:N=N+10:LT=1
490 GOTO360
500 '

******************************************************************
       FINISH UP

510 IF E THEN PRINT#1,N;"'.BL";MID$(STR$(E),2)
520 CLOSE:IF N2 THEN KILLN1$,N1:  NAME"AUTO.TMP"AS N1$,N1
530 CLEAR200:END
540 '
******************************************************************
                        <<< SUBROUTINES >>>
******************************************************************
      GET A LINE

550 Z1=Z1+1:IF FL THENGOSUB800
560 B$="":OUT17,7
570 WAIT16,1
580 X=INP(17):X=XAND127
590 IF X=6 THEN FL=(FL XOR 1):
       IF FL THENOUT17,13:OUT17,27:OUT17,75:GOSUB800:PRINTB$;
600 IFX=127 OR X=95THEN670
610 IF X=9
      THEN I=LEN(B$)MOD8:G=7-(I-1)MOD8:B$=B$+SPACE$(G):
           FOR I=1 TO G:WAIT16,2:OUT17,32:NEXT
           IF LEN(B$)>71
               THEN OUT17,7:GOTO570
               ELSE 570
620 IFX=13 OR X=10 THEN PRINT:GOTO700
630 IF X<31 THEN OUT17,7:GOTO570
640 B$=B$+CHR$(X)
650 OUT17,X:GOTO570
660 '
--------------- ECHO RUBOUT

670 F=LEN(B$):IFF<2THENIFF=0GOTO570ELSEB$="" ELSE B$=LEFT$(B$,F-1)
680 OUT17,&033:OUT17,&0104:OUT17,32:OUT17,&033:OUT17,&0104:GOTO570
690 '
---------------COUNT AND REMOVE LEADING SPACES

700 B=0
710 B=B+1:IF MID$(B$,B,1)=" " THEN 710
720 B$=MID$(B$,B):B=B-1:RETURN
730 '
******************************************************************
               BEEP AN ERROR
740 FORL=1TO500:PRINTCHR$(7);:NEXT:RETURN
750 '
******************************************************************
               APPEND TO FILE
760 OPEN"O",1,"AUTO.TMP",N1:N2=1
770 IF EOF(2)=0 THENLINEINPUT#2,A$:PRINT#1,A$:GOTO770
780 N=VAL(A$)+10
790 CLOSE2:GOTO330
795 '
******************************************************************
               PRINT COLUMN SCALE ON CRT
800 WAIT16,2:FORZ=1TO7:PRINTTAB(Z*10-1);MID$(STR$(Z),2);:NEXT:PRINT
810 Z1$=MID$(STR$(Z1),2)+" ":PRINTZ1$;MID$(Z$,LEN(Z1$)+1);
820 FORZ=1TO6:PRINTZ$;:NEXT:PRINTLEFT$(Z$,9)
830 RETURN
```

KILOBAUD KLASSROOM NO. 11

Data and Address Buses

After a slight delay, Kilobaud Klassroom is back in full swing (and will continue so until conclusion). Building your own computer is the final objective of the course. Let's do it!

Peter A. Stark
PO Box 209
Mt. Kisco NY 10549

In the last session, Kilobaud Klassroom covered the structure of the data and address buses in a typical microcomputer system, as well as some of the specialized integrated circuits that are used with them—decoders and Tri-state bus drivers.

In this session we will spend some more time talking about these buses—what they are used for—and discuss a few more ICs that come into play. You will notice that the tone of the Klassroom is changing slightly—up until now, we experimented with individual ICs and circuits, one at a time, to get an idea of just what all this digital stuff was all about.

Now we are at the point where we are slowly starting to put all these parts together into whole sections of a computer, and pretty soon we will be able to wrap up the entire system and actually start building a working computer. And it's going to be a dilly!

Introduction

Let's take a look at a diagram of a very simple computer in Fig. 1. Here we see a typical microprocessor connected to a block of memory through three sets of connections—the data bus, the address bus and a collection of control signals we might call the control bus. No input or output equipment is shown here and the amount of memory is small, so this system would not be very usable . . . but it is a start. Let's look at each of the three buses.

Most of the popular microprocessors right now work on eight-bit chunks of data called *bytes*. These bytes are stored in memory and can represent just plain numbers, letters or instructions to the computer. The data bus carries these bytes to and from the microprocessor; it is like a major highway connecting everything in the system—processor, memory, and input and output devices. It is shared by all of these devices.

Right at the microprocessor, the data bus consists of eight pins that carry data both into and out of the processor. For this reason we call the data bus a *bidirectional* bus.

But, as mentioned last time, the output circuitry in most microprocessors can only drive a light load, typically one TTL device. Although most microprocessors aren't really TTL, their output circuitry is beefed up to provide TTL signals, which involve greater currents than are normally required by the MOS circuitry (metal-oxide-semiconductor or field-effect transistor circuitry), which the rest of the microprocessor uses.

The reason for limiting the power that the microprocessor can provide is that more powerful output circuitry would require more space and power, both of which are already at a premium. As a result, we have to be careful what we connect to the microprocessor pins.

Since most memory components and other ICs used in microprocessor systems are also MOS circuits, they often provide a lighter load than even a TTL gate. Hence if a system is very small, it may be possible to connect several memory or input-output devices directly to the microprocessor bus without any additional amplifiers in the line. This is often done in small microprocessor systems that are built into other devices, and where the manufacturer wants to cut costs as much as possible.

But most hobby or experimental computers are designed for expansion to more than just a few ICs, and so additional amplifiers have to be inserted into all three buses to provide greater drive. There is usually one set of buffer amplifiers very close to the microprocessor as shown in Fig. 1, and there will often be a second set of buffers on each memory or input-output board.

In the case of the bidirectional data bus, the buffer amplifiers also have to be connected to permit data to go in both directions. There are two ways in which this can be done.

At the microprocessor, the data bus has to be bidirectional because the number of pins on the IC is limited and the eight data pins have to do double duty. But that need not necessarily be true in the rest of the system—throughout the rest of the computer we could have either a single 8-bit bidirectional bus, or two 8-bit one-directional buses.

For example, the SS-50 bus used in the SWTP computer uses a single bidirectional data bus on its 50-pin bus. The S-100 bus, on the other hand, uses two 8-bit one-directional buses—one for carrying data out to memory and the other for carrying data back. Both approaches have their good and bad points and both work, so the choice is up to each designer.

The wiring for one bit of the data bus for both of these two options is shown in Fig. 2. In both we have a set of DM8097 (or 8T97 or 74367) Tri-state buffers controlled by a $R/\overline{W}$ line from the microprocessor through a pair of inverters. When the $R/\overline{W}$ line is high, indicating a read operation, the top 8097 buffer is turned on and the bottom one is off; the opposite is true when $R/\overline{W}$ is low for a write to memory. (See Kilobaud Klassroom No. 10 in the May 1978 issue for more information on the 8097 series of buffers.) In this case, the 8097 is a non-inverting buffer; there is an 8098 that has exactly the same pin-outs, but inverts the signal.

Because there is a common need for a set of back-to-back Tri-state buffers, there is another pair of ICs that is more common than the 8097 in this application. Fig. 3 shows the pin-out for the DM8833 non-inverting quad Tri-state transceiver from National Semiconductor, which has four pairs of buffers as shown.

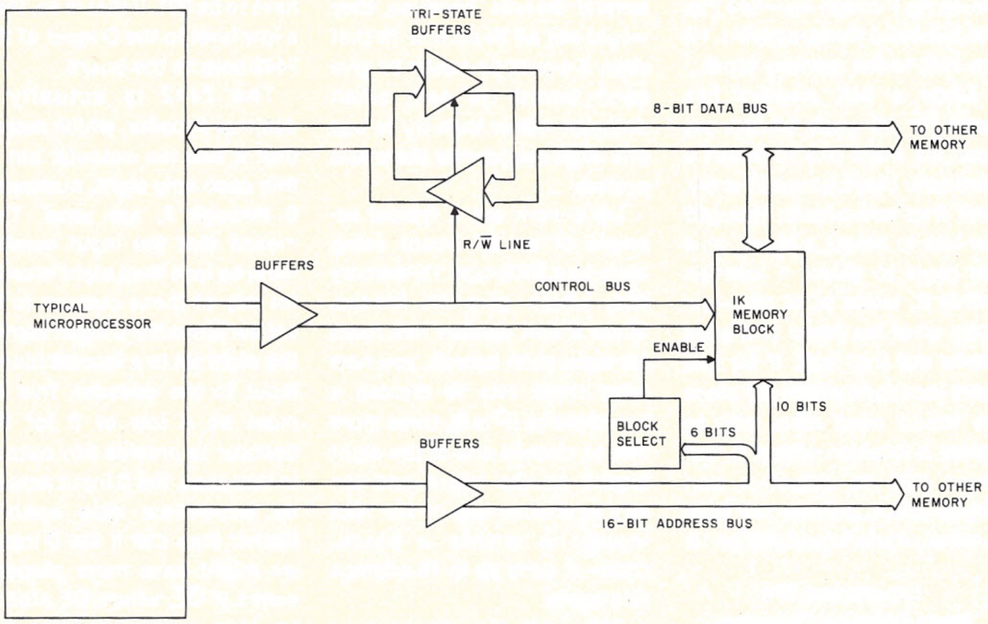


Fig. 1. Typical microprocessor bus system.

There is also an 8835 that has identical pin-outs, but is inverting. Since both of these can send data as well as receive it, they are called transceivers.

National Semiconductor makes an even better transceiver, which is more compact but also more expensive and, as yet, harder to get. This is the INS8208; as shown in Fig. 4, it has *eight* transceivers in one 20-pin package. The INS8208 is just perfect for an 8-bit bus. It has two control pins—a T/$\overline{R}$ pin that reverses direction of transmission and a $\overline{CE}$ pin that must be low to enable the chip.

The way the buffers are connected in Fig. 2, one buffer Amp of each pair is always on and the other off. This is because they are both controlled by a single R/$\overline{W}$ line that always must indicate either read or write—as is the case with, for example, the SWTP 6800 system and other processors based on the Motorola 6800.

The S-100 bus, on the other hand, has a pair of signals called MWRITE and MREAD for memory reading and writing. By simply turning off both of these at the same time, the processor can disconnect itself from the bus entirely.

OK, now let's talk about the address bus. This bus needs buffers for the same reasons as the data bus—to provide additional current drive for a greater load. But the address bus is just one-directional so Tri-state buffers are not really needed in most systems, since addresses come from just one place—the microprocessor. Still, there are cases where we want to accept addresses from other sources—perhaps another microprocessor in a multiprocessor system, or from an input-output device if we use an input-output scheme called *direct memory access*, where an I/O device communicates directly with the memory without sending data through the processor.

So the address buffers are generally the same Tri-state buffers used elsewhere. Although the hex 8097-type buffer can be used, it is much easier to spread out the required 16 buffers in two groups of eight using 81LS95 octal buffers shown in Fig. 5; some designers do it that way.

In most popular systems, the address bus uses 16 wires; since each can carry either a binary 1 or a 0, there are 2^{16}, or 65,536, locations in memory, each having its own distinct address. This means that 16 buffer amplifiers are needed. But if the system has less memory it might get by with fewer bits; for instance, if only 4K were used in some small system, then only 12 bits would have to be used and buffered. In fact, some microprocessors take advantage of this fact.

For instance, the SC/MP, or Scamp, from National Semiconductor is designed for smaller systems and only brings out 12 address pins, although it can use 16-bit addresses; the other four bits are sent out over the data bus at a time when the data bus is otherwise free. If they are needed, four flip-flop latches (using a 7475 or 74LS95 quad latch) have to be used to grab these four bits from the data bus at just the right instant. If 4K or less of memory is used, they can be ignored.

Let's look at those 16 address bits again. Although this allows addressing 64K of memory (one K is 1024, so 64K is a total of 65,536 locations), most computer systems will have less memory, and the available memory may be present in smaller chunks.

In a typical system, the

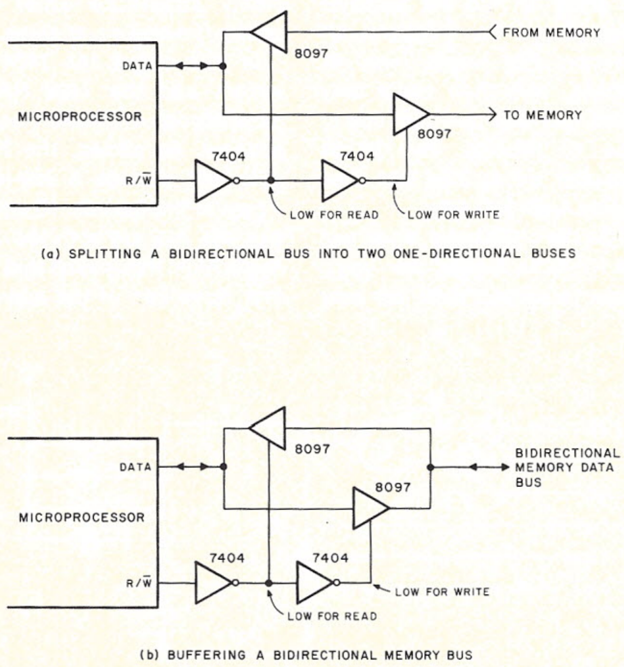


Fig. 2. Data bus buffers.

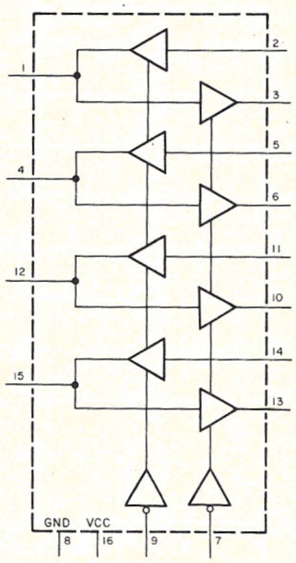


Fig. 3. Pin-out of DM8833 non-inverting quad Tri-state transceiver. When pin 7 or 9 goes low, the corresponding buffer is turned on.

memory might be divided into smaller sections—we can call them blocks—having perhaps 1K or 4K locations. Hence not all 16 bits of the address are needed to select a particular location within a block. So the 16 address bits are divided into two groups—one group is used to select a particular block of memory, and the rest select a location within that block.

For instance, suppose the memory block has 1K locations as shown in Fig. 1. Since 1K or 1024 is 2^{10}, we need ten bits to select one location within that block. And so ten bits (the ten least significant bits usually

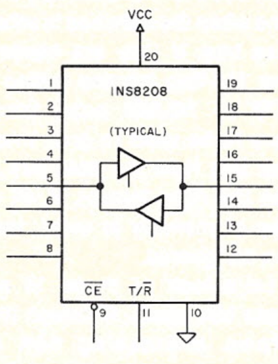


Fig. 4. Pin-out of INS8208 octal Tri-state transceiver. $\overline{CE}$ must be low to enable chip; data direction is to the right when $T/\overline{R}$ is high.

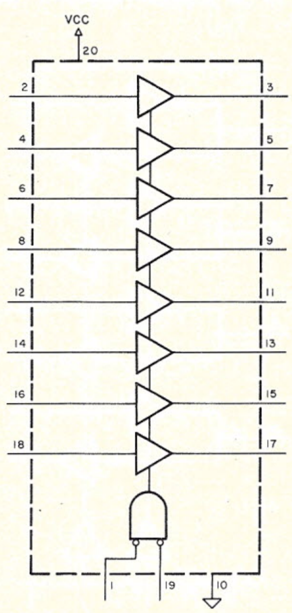


Fig. 5. 81LS95 octal Tri-state buffer. Pins 1 and 19 must both be low to transmit data; if either is high then the buffer outputs to an open circuit.

called A_0 through A_9—see Fig. 1 in Kilobaud Klassroom No. 10) are sent to the memory block.

Since a 64K memory has room for 64 such blocks, we need six more bits to select a particular block (since 2^6 is 64). Isn't that convenient, though? Six bits to select a block, plus ten more to select a location within it, totals to 16 bits—exactly what we have! It always works out that way. If we had more locations within the block we would need more bits to select a location within it, but fewer bits to select a particular block, so that the total number of bits would still be 16.

Experiment #54
More Complex Decoders

Problem: That doesn't sound very convenient at all! If you need six bits to select one of 64 blocks, what IC do you use? All the decoders we tried out last time only accept four inputs and have a maximum of 16 outputs. A decoder to accept six input bits and provide 64 outputs would need over 70 pins—isn't that a bit much?

Solution: Use several decoders or use additional gates.

Theory: Actually, this isn't as much of a problem as it sounds. If you put all 64K on one printed circuit board, then you'd have to use larger memory blocks to fit them all into the available real estate. For instance, 4K memory ICs would form 4K blocks, and then you'd only need four bits to select one of the 16 blocks. The 74154 decoder we studied last time would work just fine.

On the other hand, if you use 1K ICs in 1K blocks, then you'd have to spread them out over several boards because they would not all fit on one. In that case it would be easier to give each board its own small decoder, rather than try to use one large decoder for the whole system.

Let's work out a typical example of an 8K board using sixty-four 2102-type 1K memory ICs, which give 1K blocks. Since a 64K system has room for eight such boards, you now need three bits to choose a board (since 2^3 is 8), three more bits to choose one of the eight blocks on the board and ten bits to choose one of the 1024 locations within a block. Nice—it adds up to 16 address bits again.

Although the decoder to select a board could be placed elsewhere in the system, in practice it, too, would be on the same 8K memory board. Since it need only select one board, it needs only one output. In many cases, a gate or two would be used to do the decoding instead of using a full decoder IC.

Procedure: Fig. 6 shows a circuit that might be used on an 8K board divided into 10 blocks. For this experiment, you will need the 7442 decoder, a 7404 inverter and a 7420 dual four-input NAND, all of which you already have from previous experiments. If you don't have these specific units, you can easily substitute other decoders, inverters and gates.

In this circuit the 7404 inverters and the 7420 NAND gate decode a three-bit board address. As shown, the 7420 gate will provide a low-level signal to the decoder IC only if A_{15} and A_{14} are both low and A_{13} is high; this means that the first three bits of the address have to be 001. If that is so, then a low level to the D input of the 7442 selects the board.

The 7442 is actually a decimal decoder, which means that it has ten outputs; but in this case only eight, numbered 0 through 7, are used. This means that the input number must be less than eight in order for any of these outputs to be active. Since the D input carries the eight bit in a four-bit binary input to the decoder, this bit must be a low or 0 for the decoder to provide any output. In this case, the D input acts as an active-low chip enable. Rather than call it the D input, we might have more properly called it $\overline{CE}$, where CE stands for chip enable, and the bar over it signifies that it must be low.

The next three bits, A_{12} through A_{10}, provide the other three inputs to the 7442 and select one of the eight outputs. In this experiment, these outputs light one of eight LEDs, but on an actual memory board they would select one of the eight 1K memory blocks.

Since the first three bits must be 001 as wired, this board would respond to hexadecimal addresses 2000 through 3FFF. But this could be changed by simply changing the inverters around. For example, if all three inputs into the 7420 gate had inverters, then only addresses beginning with 000 would select the board, and the board would then have addresses 0000_{16} through $1FFF_{16}$.

On the other hand, if no inverters were used, then all addresses would have to start with 111. If several 8K boards were used in a system, each one would have to have a different combination of inverters so that each would have a different set of addresses. This might get somewhat inconvenient, but is an OK approach for just one board.

There are several ways to eliminate the inverters and 7420 gate. Fig. 7 shows how two 7442 inverters do the same job. If the top three bits are 001, then the top decoder sends a low signal to the second one,

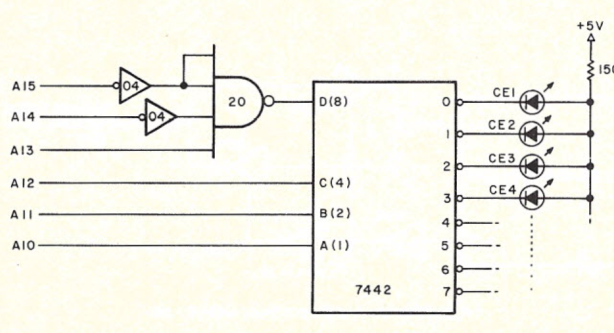


Fig. 6. A more complex memory-select circuit.

which then selects a particular block of memory depending on the next three bits. Moving the entire board to a different memory address area can be done just by changing the jumper between the two decoders.

A better choice is to use a decoder specially designed for memory decoding, the 74LS138 shown in Fig. 8. Not only is this IC faster than two 7442s, but also the addition of three chip enable pins makes it possible to do some board selection without the use of additional gates or inverters. In this case, we connect the top two address bits to the $\overline{CE}$ inputs and the next bit to the CE so the addresses again have to start with 001. This decoder is fairly popular in some of the hobby-computer memory boards.

Experiment #55
Partial Address Decoding

Problem: What if the number of inverters or the $\overline{CE}$ and CE lines don't happen to fit the high/low combinations we have?

Solution: Take a shortcut.

Theory: Suppose we were using the 74LS138 decoder but had to decode a board address of 101 instead of 001. This would require testing for two highs, not just one, so we would need two CE inputs and only one $\overline{CE}$ input. We could, of course, use an inverter to flip A_{15} upside down, but there may be an easier way.

Procedure: Take the circuit of Experiment #54—it doesn't matter whether you used a 7442 and some gates and inverters, two 7442s or a 74LS138—disconnect the A_{15} line from the decoder, and instead ground this input. Of course, on the experimenter's console you are using the new circuit doesn't look much different from the old, since you were using wires to provide high and low signals to the decoder before, too; and so the new circuit grounding that input looks just like the old circuit grounding the same input. At this point, you have to apply a little imagination.

The point is that before we had the decoder looking at the

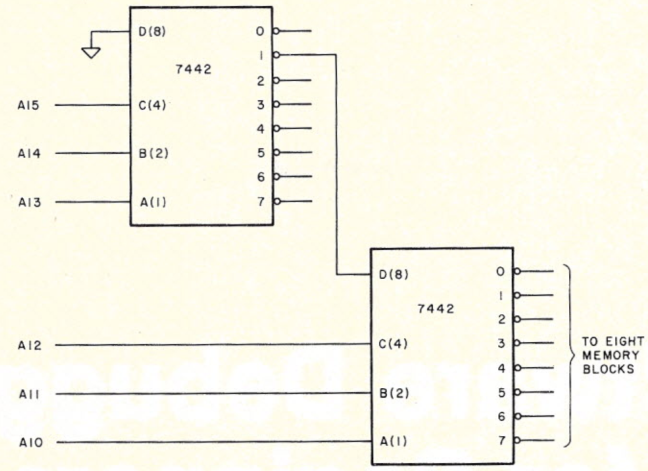


Fig. 7. Using two 7442 decoders for complete address decoding.

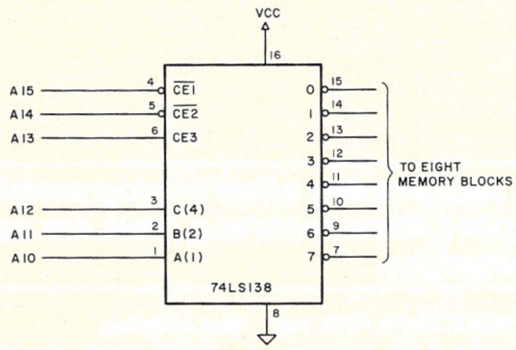


Fig. 8. The 74LS138 decoder is specially designed for memory decoding.

most significant three bits of the address bus—A_{15}, A_{14} and A_{13}—and checking for a 001 code. Now we are looking only at A_{14} and A_{13}, and don't care what A_{15} is. Using the letter X for a "don't care" signal that could be either 0 or 1, the first three bits now have to be X01. That is, this memory board will now react not only to addresses starting with 001, but also to addresses starting with 101.

For instance, if we store something onto memory location 0010001100110011 (address 2333 in hexadecimal), we can later read it out of either the same location or out of address 1010001100110011 (or hex address A333). As far as the memory is concerned, location 2333 and A333 are the same. This applies to every other memory location on this board—each location now can be accessed with two different addresses.

Decoding only part of an address like this is called *partial* or *incomplete address decoding*. It simplifies the circuitry, and is often used for that reason. In this particular case we just omitted one address bit from the decoder circuit, but often the designer of a system will omit several bits. Every time you eliminate one bit from the address decoder, you double the number of addresses assigned to the same location.

Incomplete address decoding has, of course, some disadvantages as well. In the decoder circuit we discussed in experiment #54, we talked about a memory board having 8K of memory addressed from 2000 hex to 3FFF. With the incomplete decoding—skipping bit A_{15}—this same board also has addresses ranging from A000 through BFFF. In other words, we are using up 16K worth of addresses for only 8K of memory. As long as we don't need those 8K addresses at A000 and up, this is OK. But if at some later time we decide to expand the system up to its full memory potential of 64K, we will have trouble because we will not be able to use those addresses.

Incomplete address decoding is commonly used in special-purpose microcomputers, such as those that might be used as controllers inside a cash register or inside a test instrument, where saving some room or money becomes important. It is not a good idea to design it into a general-purpose system that may be expanded in the future. Even so, you still see it in a few systems designed back in the days when memory was expensive and people thought a system was big when it had 8K of memory.

Preview

This finishes the subject of the address and data buses, and some of the special ICs used with them. We will leave the control bus until later, since the details of what lines are available on this bus differ greatly from one microprocessor to the next.

Next month we will look at RAM and ROM memory ICs, and at the circuitry on a typical memory board. For our experiments we will need about half a dozen inexpensive signal diodes (they don't have to be very good since we are going to purposely burn out a couple of them, so get them out of your junk box), and a 2102, 21L02, or 2102L or similar 1K static memory IC. These ICs are available by mail order for about $1.25 from various sources, but for just one it is probably easier to buy it from your local Radio Shack (part number 276-2501).

One last comment—this IC is a MOS circuit, which is easily damaged by static electricity. When you get it, resist the temptation to take it out of its package, unless you know how to handle MOS ICs. We will give you the precautions for handling these ICs next month, just before we use the chip in the next experiment. See you then! ■

Software Debugging for Beginners

Sometimes it must seem as though every effort in programming is one big error message. Aside from reading the supplied documentation, here are some useful tips that you can follow for avoiding those messages.

John Leslie
10 Souhegan St.
Milford NH 03055

If it hasn't happened to you yet, it has surely happened to someone in your computer club: After having sent your check for a $15 program, you received the program, loaded it, and it didn't work. Then, of course, you called the people who sold you the program to tell them the sad story. In general, people selling software to computer hobbyists do try to be helpful.

Often, though, you can have serious difficulty finding someone familiar enough with the program to help you. And usually, you end up waiting several weeks for a replacement cassette. Most people find this frustrating. Why can't you do something in the meantime?

Well, I am about to tell you that you can. You might get scared if I told you to debug the program, so, instead, I shall suggest simply that you play around with it. But in truth, you will be doing much the same things that a professional programmer does when he debugs.

Study the Assembly Listing

To start, you should read through the assembly listing, with both source code and object code. Usually this listing is supplied with the program. Oops, I think I lost a few of you there.

An *assembly listing* is simply a listing (printout) produced by an assembler. *Source code* is what the programmer actually wrote, and is used as input to the assembler. *Object code* is a representation (usually in hexadecimal) of the actual bit patterns the computer will use as its instructions.

An *assembler* is a program or process that puts together a machine-language program, given a *source program* specifying each machine-language instruction in detail. Many of the programs supplied to hobbyists have been *hand assembled,* meaning simply that the programmer, rather than the computer, assembled the program. It's of no consequence. He will have produced the same sort of listing.

Don't worry about understanding the listing in detail; don't even worry about understanding it in general. Your task is just to gain some familiarity with it.

First, Read the Comments

What do you look for? Look at the pictures. Sad to say, there usually aren't any. Next, look at the comments. Assembly code has the following format:

Label / Op code / Operands / Comments

Comments are off to the right, and are the only things that remotely resemble English. If they are written well, comments will tell you *why* something is done, rather than *what* is done. And at this stage, you aren't much concerned with *what* is being done.

The sample assembly listing (Fig. 1) is an actual portion of a program of mine. It performs a generation change in the game of Life. Reading the comments, you should get the idea that the program tests something about eight neighbors, and sets a cell's new state on the basis of its previous state and the number of its neighbors. That is enough to grasp for now.

Then, the Jump Op Codes

Next, look at the op code column. You should have a manual explaining the various op codes for your computer. You may have tried reading it, and given up. But don't worry—it is not meant to be read; it is a reference book. So now, get it out and *refer* to it. You want to know which of those alphabet-soup concoctions called op codes are supposed to cause the computer to jump to a new location. Memorize them by name. Don't worry too much about what they do.

In Fig. 2, I have outlined the jump op codes for the 8080, 6800 and 6502 microcomputers. This should enable you to get through the example of Fig. 1 if you don't have a 6502.

The point of this is that all the other instructions will execute without a break in sequence, so you can safely ignore them; whatever they may do, the computer will proceed to the next instruction in sequence.

The jump op codes, on the other hand, can bring untold

confusion. They can cause machine instructions to be executed many times, or not at all.

Outline the Program Flow

So your next step is to scan the op code column for these jump op codes. This longer exercise is going to give you a picture of the *program flow.*

In the example of Fig. 1, you will find eleven uses of JSR, three uses of BNE and one each of BVC, BVS and RTS. You should assume that all subroutines called will usually return to the next instruction. (This is not necessarily true, of course; sometimes it is spectacularly false. But you should assume it anyway.)

So ignore all those JSRs. The three BNEs turn out to simply skip the next instruction, so you can ignore the first two. The third is a conditional skip of the return from subroutine instruction, so you will have to watch it.

The BVC and BVS instructions are both conditional jumps, and if you check the op code table of Fig. 2, you will see that they are opposite conditions. Thus, these two instructions together amount to an unconditional jump back to near the beginning of the example. (Using two conditional jumps like this is common in 6502 code, since there is no relative-addressed unconditional jump instruction.)

You now have a picture of the program flow . . . straight through until near the bottom, where there is a conditional skip followed by a return from subroutine instruction (this amounts to a conditional return instruction). If the return is not done, then most of the code is repeated.

A few inferences are in order. If we execute a return from subroutine instruction, this whole thing must have been called as a subroutine. (True.) And eventually the conditional skip must fail, and the return will be executed. (Also true.)

A Word about Operands

Now, I suppose I should confess to sneaking a few things by while you weren't looking. I promised that you would be looking at the op codes column, but I went right ahead and checked the next column (operands) whenever I felt like it. And since I haven't told you anything about that column, you feel justifiably confused.

I approach this section with trepidation, fearing that after I have told you about the operands column you will be even more confused. The plain fact is that this column is used for all the leftovers. The theory is that "obviously" some additional information is needed to complete the machine instruction, and it is "simply" placed in the operands column.

The experienced programmer never gives this a second thought. Depending on his background, he may think of the items in that column as addresses or data. The fact remains, however, that they are leftovers, and no consistent rule can be made to apply. If you attack this slowly, however, and memorize what is put in that column for each of several kinds of op codes, you will soon enough master it sufficiently.

For right now, you only need to know about the jump op codes. For these, obviously the needed information is where to jump. So, keeping in mind that whatever you see must be a description of where to jump, look over the entries in the operands column following the jump op codes. You will see BEGIN, TEST, SWAP, GLOOP and $+2. "What's this $+2 bit?" you may well ask.

Permit me to digress a moment. If somebody asks for directions, you probably will use your hand to point. "Over there," you say, but your pointing is what conveyed the information. Computers are not too good at reading hand motions. The computer requires exact latitude and longitude, so to speak. People, on the other hand, much prefer to say, "over there," and point. Clearly, some compromise needs to be reached. Several methods have been tried over the years, with two still being used frequently.

```
       Label                         Op Code  Operands        Comments

0328  20 1C 02     GENERATION   JSR   BEGIN         Preset to XMIN, YMIN
   B  A6 D3        GLOOP        LDX   CURRX         X coordinate
   D  A4 D4                     LDY   CURRY         Y coordinate
   F  A9 00                     LDA   =0
  31  20 14 02                  JSR   TEST
   4  85 D2                     STA   CURR          1 iff occupied
   6  A9 00                     LDA   =0            Preset no neighbors
   8  E8                        INX                 1st neighbor
   9  20 14 02                  JSR   TEST          Count if occupied
   C  C8                        INY                 2nd
   D  20 14 02                  JSR   TEST
  40  CA                        DEX                 3rd
   1  20 14 02                  JSR   TEST
   4  CA                        DEX                 4th
   5  20 14 02                  JSR   TEST
   8  88                        DEY                 5th
   9  20 14 02                  JSR   TEST
   C  88                        DEY                 6th
   D  20 14 02                  JSR   TEST
  50  E8                        INX                 7th
   1  20 14 02                  JSR   TEST
   4  E8                        INX                 8th
   5  20 14 02                  JSR   TEST
   8  A0 00                     LDY   =0            Preset dead
   A  C9 02                     CMP   =2            If two neighbors
   C  D0 02                     BNE   $+2
   E  A4 D2                     LDY   CURR          Survives
  60  C9 03                     CMP   =3            If three
   2  D0 02                     BNE   $+2
   4  A0 01                     LDY   =1            Grows
   6  20 44 02                  JSR   SWAP          Work on new board
   9  98                        TYA                 New state
   A  A6 D3                     LDX   CURRX
   C  A4 D4                     LDY   CURRY
   E  20 18 02                  JSR   SET           Store new state
  71  20 20 02                  JSR   STEP          Advance to next cell
   4  D0 01                     BNE   $+2
   6  60                        RTS                 Finished board
   7  20 44 02                  JSR   SWAP          Back to old board
   A  50 AF                     BVC   GLOOP
   C  70 AD                     BVS   GLOOP
```

Fig. 1. Sample assembly listing.

The first is to write notes to the computer saying, in effect, "This is the place I call home; this is the place I call school; and this is the place I call work." Later on, the computer, being suitably programmed, will know what to do when you tell it to go to school.

The second method saves time when you don't feel like writing a note giving someplace a name. It is the equivalent of saying, "second door on your right." $+2 is analogous to second door on your right. Be warned that this kind of notation is not the same for all assembly languages. Some use a dollar sign; some use an asterisk; some use a number sign; and goodness only knows how many different characters have been used for the function.

In every case, however, this special character refers to the current location. But, sad to say, not every assembler means the same thing by current location. Some mean the address of the machine instruction being assembled; some mean the contents of the internal program counter register when the instruction is executed (which usually points to the instruction after it).

And, as if that weren't bad enough, not all assemblers count the same thing when determining what +2 means. Some count instructions; some count bytes; assemblers for larger computers may even count words or half-words. But at least the plus vs minus directions are standard. Plus refers to higher addresses (instructions later in sequence) and minus refers to lower addresses (instructions earlier in sequence).

"How," you may ask, "do I tell what it means?" Well, if you're lucky, you may not have to. If you see $+2 and neither of the next two instructions is a jump, you can ignore the problem for a while. When you finally are forced to find out for sure, I recommend running the computer in single-step mode and finding out what *it* thinks. That, after all, is the only opinion that

is of any significance.

One of those manuals you received with your computer will, no doubt, tell you what the standard is, according to the manufacturer. That is the standard you should follow; and for any code run through the assembler, you know it is the convention the assembler followed.

But, as I mentioned before, many programs supplied to computer hobbyists have been hand assembled, and you *don't* know what convention that programmer may have used. Programmers have a tendency to use whatever scheme they were brought up with, until forced to change. Until you are quite certain which scheme the programmer used, don't assume anything. And, in case of doubt, check what the computer thinks.

On my way back to the subject, let me explain the first scheme—writing notes. The label field, starting with the first character of each line, is reserved for notes assigning names to places. On the second line of Fig. 1, you see a note: GLOOP. If there is anything in the label field, that name is assigned to that program location. Thus, the name GLOOP is assigned to program location 032B. When, near the end, you see BVC GLOOP, that causes a conditional jump to 032B to be assembled.

Now we can get back to business. I have told you about the operands for jump op codes, and we can get back to analyzing program flow. "Why," you may well ask, "make such a fuss about program flow?" The answer is quite simple, really. It makes no difference whether code is correct if it is not being executed . . . and, I might add, being executed the right number of times.

6800	6502	8080	Jump type
*BRA	JMP	JMP	Unconditional
JMP	---	---	Indexed
BNE	BNE	*JNZ	If not equal
BEQ	BEQ	*JZ	If equal
BCC	BCC	*JNC	If carry clear
BCS	BCS	*JC	If carry set
BPL	BPL	*JP	If plus
BMI	BMI	*JM	If minus
BGE	---	---	If greater than or equal
BGT	---	---	If greater than
BHI	---	---	If higher
BLE	---	---	If less than or equal
BLE	---	---	If lower or same
BLT	---	---	If less than
BVC	BVC	---	If overflow clear
BVS	BVS	---	If overflow set
---	---	JPO	If parity odd
---	---	JPE	If parity even
---	---	PCHL	Load PC from H,L pair
*JSR	JSR	CALL	Subroutine jump
---	---	CNZ	Conditional subr jumps
---	---	CZ	
---	---	CNC	
---	---	CC	
---	---	CP	
---	---	CM	
---	---	CPO	
---	---	CPE	
**SWI	BRK	**RST	Software interrupt
**WAI	---	**HLT	Wait for interrupt / Halt
RTI	---	---	Return from interrupt
RTS	RTS	RET	Return from subroutine
---	---	RNZ	Conditional returns
---	---	RZ	
---	---	RNC	
---	---	RC	
---	---	RP	
---	---	RM	
---	---	RPO	
---	---	RPE	

Fig. 2. Op codes on the same line are reasonably compatible. A single asterisk indicates a different addressing mode. A double asterisk indicates gross detail incompatibility (but the functions are similar).

Set Breakpoints

Consequently, before charging off to check what a piece of code does, you should set a breakpoint to make sure it is being called. Very often, it isn't.

Some computer systems have powerful and easy-to-use methods for setting breakpoints. But on your microcomputer, it is probably pretty cumbersome. If someone knowledgeable is nearby, by all means ask him what the easiest way is. Most likely, it will involve overwriting the location where you want a breakpoint with an instruction to cause a software interrupt (BRK for the 6502, SWI for the 6800, or RST for the 8080), and assigning a monitor routine to field the interrupt.

To resume after the breakpoint, you will probably have to restore the instruction you overwrote, reset the program counter there and possibly fix the stack. If this sounds like a lot of work, you now understand why you keep seeing articles about better monitor systems. Nonetheless, I assure you, it is worth the effort.

Having a picture of the program flow, and knowing how to set breakpoints, you can now start setting breakpoints all over the place to prove whether the program is actually being executed according to your picture of the program flow. Surprisingly often, something has gotten garbled along the way, and the computer turns out to be jumping into some strange area. If you find such a case, it is usually easy to fix.

Your first hint of this is usually that the computer never reaches a breakpoint that you have set. Then proceed to set breakpoints gradually earlier (restarting the program each time), until the computer does stop. By this method, you can pinpoint where the program goes astray. Comparing the machine-language instruction (in memory) to the assembly code will quickly show any case of garbling.

This Program Will Self-destruct . . .

You should be warned that sometimes when a program goes haywire, it overwrites itself. You should always reload the program from cassette (or whatever medium you use) before setting a new breakpoint and restarting. However, this is *so* much of a nuisance that experienced programmers seldom do it. Nonetheless, when you're correcting a garbled instruction, it is worth the effort to reload and check whether it was garbled as loaded. If not, you haven't found the problem yet.

If you find a case in which the program is clobbering itself, you should set breakpoints progressively earlier, checking each time to see whether it has clobbered itself yet. For this case, of course, it is necessary to fix the clobbered code, usually by reloading from cassette before each restart.

Each time you find and fix a bug, you should feel free to remove all breakpoints to see if the program as a whole now works. If, on the other hand, you debug a newly written program, you should pause and spend a few minutes looking for similar mistakes. The human mind, once it has made a mistake, tends to make it again. It may even be worth your while to scan for similar mistakes when debugging a program that once worked.

Examine Variables

After you have proven that the program is being executed according to your understanding of program flow, it makes some sense to look at the program variables to see if they contain reasonable values at the strategic points during the computation.

As an example, it is often helpful to check the value of index variables at the beginning and end of iterative loops to see whether the loop is being done the right number of times. In the example of Fig. 1, CURRX and CURRY are index variables that represent the X and Y coordinates. Thus, you might reasonably expect them to range from 1 to 40 and from 1 to 24. Typically, at the end of iteration, one index variable will be

at the final value or one past it. The rest should all be at final value or at initial value. If not, you have grounds for suspicion.

If you become suspicious, you can execute the loop exhaustively and count the number of times it is done. If, on the other hand, an index variable gets an obviously wrong value, you should suspect it is being clobbered and proceed to test where it is being clobbered.

It is also instructive to examine data areas during sections of code that are not supposed to change them, to ensure that they are, in fact, not being changed. If they are being changed, you can use the standard procedure to zero in on where the changes are occurring.

Data Structures

If you are particularly lucky and the program was well designed, there will be a subroutine you can call to display the status of data areas. When writing your own programs, you should be sure to include such a subroutine, preferably in a form that changes *nothing*, so that it may be called between any two instructions during the debugging phase.

If you are that lucky, you can now run through the section of code that is supposed to modify the data, setting breakpoints at convenient locations, and examine the data as changes are being made. Using this feature, you can often pinpoint the trouble area, still without having to know *what* the code is doing. But more likely, you won't be that lucky, and you will have to set out to learn about the data structure. And that, I fear, must wait for another article.

Summary

In debugging, you should always first establish which code is being executed. Then check to see that loops are being done the right number of times. After the program flow is proven correct, check that variables contain reasonable values. Only after you have localized a problem do you set out to understand what the code is doing.

Postscript—BASIC

To debug BASIC programs, you usually insert PRINT statements. In keeping with the debugging principles listed above, your first task is to establish program flow. So insert the simplest possible PRINT statements, using them like breakpoints. After program flow is established, then you should switch to PRINTing the values of the data. Printing your data before establishing program flow leads to much head-scratching and little progress. ■

Fred R. Ruckdeschel
773 John Glenn Blvd.
Webster NY 14580

Mits vs North Star: Which Is Faster?

This assessment may not necessarily concur with yours, but read on for the conclusion.

There have been several earlier comparisons of speed capabilities of various microcomputers and their associated software, mainly BASIC interpreters. A fairly complete data compilation was presented in the October 1977 issue of *Kilobaud* magazine (Rugg and Feldman's "BASIC Timing Comparisons," p. 20). The programs used for generating these data intentionally did not include benchmark routines containing arithmetic-intensive operations, presumably because most current small-system applications using BASIC are string (games, word processing, etc.) oriented. However, there are a few of us in the microcomputer world who have forsaken math tables and the like and are grinding out numerical integrations, series expansions and statistical analyses with processing times sensitive to how long it takes to execute a multiplication or a division.

In the following discussion, I will show that although previous speed rankings using string-intensive benchmark programs put Mits Extended (12K) BASIC in the same ball park (though a bit faster) as North Star BASIC (both with and without a floating-point multiply/divide board), such is not the case when math operations are considered.

Benchmark Programs

The test loops used are shown in Listing 1. Each loop is cycled 1000 times. A CL2400 real-time clock (RTC), manufactured by Cañada Systems, Inc., keeps track of the computer time. The time at the beginning and end of each test loop is printed out on a CRT using a time subroutine that reads the clock. Printing time was negligible as a Processor Technology VDM-1 video display running at its maximum speed was employed. The microcomputer utilized for the timing comparisons was a standard Imsai 8080 with Seals 8K memory boards. These boards contained 250 ns static memory chips; there were no memory wait states.

A sample printout (300 baud hard-copy terminal) is shown in Listing 2. The difference between the two printed times for each loop is a result of the sum of the 1000-cycle loop-execution period, the clock-time acquisition delay and the printout time span. In the case of Listing 2 (300 baud), about one second of the time increment is due to printing. About 0.2 seconds is associated with clock reading/printout using the video display.

The first test in Listing 2 is a simple loop meant to approximately determine the time overhead above that required to interpret and perform the function calculation. The subsequent loop-time increments correspond to cases in which the assignment statement (X = J) has been changed to a calculation.

Test Loop	**1000 Cycle Time (seconds)***		
	Mits	*North Star*	*North Star FP*
Simple Assignment	4	4 (5)	4
$X = {}^{1}/J$	11	20 (21)	6
X = J*J	8	9 (11)	6
$X = J^{(10.9)}$	59	171 (223)	22
X = SIN (J)	27	103 (127)	15
X = COS (J)	27	103 (128)	15
X = SQRT (J)**	50	96 (110)	8
X = LOG (J)	23	103 (129)	13
X = EXP (J/100)	39	93 (120)	14
Mixed	164	525 (648)	64

*Results shown in parentheses are from Listing 2. That data has an extra one-second duration because of printing. Also, there is a CPU inefficiency due to the wait states associated with dynamic memory.

**X = SQR(J) for Mits BASIC.

Table 1. Comparison of North Star Floating Point BASIC with North Star BASIC Version 6, Release 2, and Mits Extended (12K) BASIC, Version 3.2; uncorrected loop times.

Results

The raw data (video display case) is shown in Table 1. It is apparent that the sine and cosine routines are equivalent. A simple trigonometric identity is probably used to relate sine and cosine, and it is likely that only one time-consuming series-expansion routine is called on for calculation (after range reduction).

It is also apparent that for all three BASICs examined the

software overhead associated with interpreting and looping is approximately the same. This is consistent with previous comparisons. Also, the absolute values (~2.5 milliseconds for a simple loop) agree with previous data. The data obtained from a different Imsai containing dynamic memory boards (otherwise presumably identical) suggest that particular machine was running roughly 20 percent slower than the reference computer. Thus, there is some danger in comparing software speeds on varying hardware and drawing conclusions based on 20 percent variances.

To isolate the times associated with interpreting and executing the function calculations, subtract the simple assignment loop time (see Table 2).

As expected, simple multiplication is the fastest calculation. (Obviously, addition and subtraction are typically much faster functions and verge on being insignificant in terms of their time consumption in an interpreter.) In this respect, the speed times for both Mits and regular North Star BASIC are roughly equivalent, with Mits a little faster; multiplication software routines are reasonably straightforward. The slight difference is probably related to the 8-digit accuracy in North Star BASIC, while Mits BASIC has 7-digit accuracy. The hardware floating-point board increases the multiplication execution speed roughly by a factor of two.

The next fastest operation is simple division. In this case, Mits BASIC is approximately twice as fast as regular North Star BASIC. However, the floating-point board BASIC turns the tables and executes divisions roughly three times faster than Mits BASIC, again with greater accuracy of one digit.

Mits BASIC has the biggest function-speed edge over North Star BASIC in the logarithm operation: Mits is five times faster. However, when compared to the North Star Floating Point Board it is twice as slow. On the average, for arithmetic-intensive calculations, Mits BASIC appears to run about three times faster than regular North Star BASIC. Also, on the average, North Star BASIC with the hardware floating-point board runs about three times faster than Mits BASIC.

Interestingly, North Star has advertised that their floating-point board increases processing speed *up to* tenfold. This *maximum* is a little conservative considering the games that are played in advertising.

Floating-Point Board Speed

Before proceeding, I should

```
RUN

LOOP TIME TEST
TIME -- 8: 11: 17
TIME -- 8: 11: 22
*********************
1/J TEST
TIME -- 8: 11: 24
TIME -- 8: 11: 45
*********************
J*J TEST
TIME -- 8: 11: 48
TIME -- 8: 11: 59
*********************
J↑10.9 TEST
TIME -- 8: 12: 2
TIME -- 8: 15: 45
*********************
SIN(J) TEST
TIME -- 8: 15: 47
TIME -- 8: 17: 54
*********************
COS(J) TEST
TIME -- 8: 17: 57
TIME -- 8: 20: 5
*********************
SQRT(J) ROUTINE
TIME -- 8: 20: 7
TIME -- 8: 21: 57
*********************
LOG(J) TEST
TIME -- 8: 21: 59
TIME -- 8: 24: 8
*********************
EXP(J/100) TEST
TIME -- 8: 24: 10
TIME -- 8: 26: 10
*********************
MIXED TEST
TIME -- 8: 26: 12
TIME -- 8: 37: 0
READY
```

Listing 2. Sample printout for regular North Star BASIC using a 300 baud hard-copy terminal with an Imsai 8080 containing Mits 4K dynamic memory boards. About one second of the time difference between the start and end loop-time printouts is attributable to the printer delay. There is also an inefficiency due to memory refresh cycles. All other tests were made with a "fast" Imsai.

```
10 REM FLOATING POINT TEST ROUTINE
20 PRINT "LOOP TIME TEST"
30 GOSUB 720
40 FOR J=1 TO 1000
50 X=J
60 NEXT J
70 GOSUB 730
80 PRINT "*********************"
90 PRINT "1/J TEST"
100 GOSUB 720
110 FOR J=1 TO 1000
120 X=1/J
130 NEXT J
140 GOSUB 720
150 PRINT "*********************"
160 PRINT "J*J TEST"
170 GOSUB 720
180 FOR J=1 TO 1000
190 X=J*J
200 NEXT J
210 GOSUB 720
220 PRINT "*********************"
230 PRINT "J↑10.9 TEST"
240 GOSUB 720
250 FOR J=1 TO 1000
260 X=J↑10.9
270 NEXT J
280 GOSUB 720
290 PRINT "*********************"
300 PRINT "SIN(J) TEST"
310 GOSUB 720
320 FOR J=1 TO 1000
330 X=SIN(J)
340 NEXT J
350 GOSUB 720
360 PRINT "*********************"
370 PRINT "COS(J) TEST"
380 GOSUB 720
390 FOR J=1 TO 1000
400 X=COS(J)
410 NEXT J
420 GOSUB 720
430 PRINT "*********************"
440 PRINT "SQRT(J) ROUTINE"
450 GOSUB 720
460 FOR J=1 TO 1000
470 X=SQRT(J)
480 NEXT J
490 GOSUB 720
500 PRINT "*********************"
510 PRINT "LOG(J) TEST"
520 GOSUB 720
530 FOR J=1 TO 1000
540 X=LOG(J)
550 NEXT J
560 GOSUB 720
570 PRINT "*********************"
580 PRINT "EXP(J/100) TEST"
590 GOSUB 720
600 FOR J=1 TO 1000
610 X=EXP(J/100)
620 NEXT J
630 GOSUB 720
640 PRINT "*********************"
650 PRINT "MIXED TEST"
660 GOSUB 720
670 FOR J=1 TO 1000
680 X=.7*SIN(J/10)↑4+LOG(2+COS(J))*SQRT(EXP(4))
690 NEXT J
700 GOSUB 720
710 END
720 REM TIME SUBROUTINE
730 FOR I=0 TO 7
740 LET T(I)=INP(168+I)
750 NEXT I
760 LET H=10*T(7)+T(6)
770 LET M=10*T(5)+T(1)
780 LET S=10*T(2)+T(3)
790 PRINT "TIME --",H,":",M,":",S
800 RETURN
READY
```

Listing 1. Test program. Note that in line 470 SQRT must be changed to SQR for Mits BASIC.

Function	Function Time (milliseconds)			Speed (1/time) Ratio		
	Mits	*North Star*	*North Star FP*	*Mits/NS*	*FP/NS*	*FP/Mits*
Division	7	16	2	2.3	8.0	3.5
Multiplication	4	5	2	1.2	2.5	2.0
Power	55	167	18	3.0	9.3	3.1
Sin/Cos	23	99	11	4.3	9.0	2.1
Square Root	46	92	4	2.0	23.0	11.5
Logarithm	19	99	9	5.2	11.0	2.1
Exponent*	28	73	8	2.6	9.1	3.5
Mixed	160	521	60	3.3	8.7	2.7

*Division in exponent (Table 1) allowed for.

Table 2. Incremental time to do function.

Composing Functions	Time (Milliseconds)		
	Mits	*North Star*	*North Star FP*
two multiplications	8	10	4
one division	7	16	2
two sine/cosines	46	198	22
one power	55	167	18
one square root	46	92	4
one logarithm	19	99	9
one exponent	28	73	8
Total (Calculated)	209	655	67
Total (Measured)	(160)	(521)	(60)

Table 3. Comparison of calculated time for mixed expression with measured time.

note that the speeds given above are based on those *perceived* by a user of the particular software package, given those *particular* arguments for those *particular* functions (and that *particular* machine). Execution time is somewhat argument sensitive, though the 1000-fold span in argument range (1⩽J⩽1000) should average that variation out.

If you were programming in machine language, the apparent execution times for the floating-point board would be shorter. For example, according to North Star data sheets on the floating-point board, an 8-digit-precision multiplication (not counting data transmission times) takes between five and 382 microseconds, with 80 microseconds being typical. Similarly, 8-digit precision division is stated to take between seven and 340 microseconds, with 156 microseconds being typical.

The multiplication and division times observed in Table 2 are more than an order of magnitude greater than the intrinsic capabilities advertised for the floating-point hardware. Presumably, the difference is due to the software overhead required for implementation. What counts for the present evaluation is the *perceived* (through BASIC) processing speed.

Function Choices

In programming mathematical routines you often have a choice between functions for achieving desired results. For example, if you wanted to square a number, should a multiplication, power or logarithm/exponent routine be used? The answer can be derived from Table 2.

To do this we will assume that execution times can be simply (in approximation) added. To test this assumption, we can estimate how long the mixed calculation should have taken given the individual processing times for the functions comprising that calculation (see Table 3). From Table 3 it is apparent that simple incremental time addition is a reasonable approach to estimating the total calculation time, with the error being on the high side.

Back to squaring a number (in Mits BASIC). One approach would be to use the power function. This would take 55 milliseconds. Alternately, you could take the logarithm of the number, double it (multiply) and then exponentiate. This could take 52 milliseconds, which is curiously similar in value to the power function speed.

The best way is to multiply the number by itself. That consumes only four milliseconds. In fact, straight multiplication is the best route to go for integral powers up to about 12. (Judicious use of parentheses can significantly expand this limit.) Nonintegral powers force you to use the other functions.

Note that in each of the three BASICs, the power function appears to be implemented through the logarithm and exponential functions (see Table 4). The power function is thus simply a convenient secondary function.

Rapid convergence algorithms for finding the square root exist. Therefore, it is expected that this function would *not* be implemented using the log/exp combination. This appears to be the case with North Star BASIC. Mits BASIC, however, *may* (this is a risky conclusion) be implementing the square-root function through its relatively fast logarithm and exponent routines.

Conclusions

For arithmetic-oriented applications it is apparent that the North Star BASIC/Floating-Point Board combination significantly exceeds the speed of Mits Extended (12K) BASIC. For string-oriented applications, the speeds are comparable.

A possible objection to using the floating-point board is price: $259 kit, model FP-A. However, along with this hardware you also get North Star BASIC software, which is comparable to Mits Extended BASIC, though it lacks some features. If you purchase the Mits software separately, the price differential between the purchased Mits software and North Star software plus hardware package is not great. Thus it might be argued that the North Star Floating-Point Board system might be preferable to Mits BASIC, particularly when arithmetic applications are considered.

In summary, if you wish to crunch numbers on an S-100 bus, the North Star Floating Point BASIC software/hardware combination appears to be the way to go.■

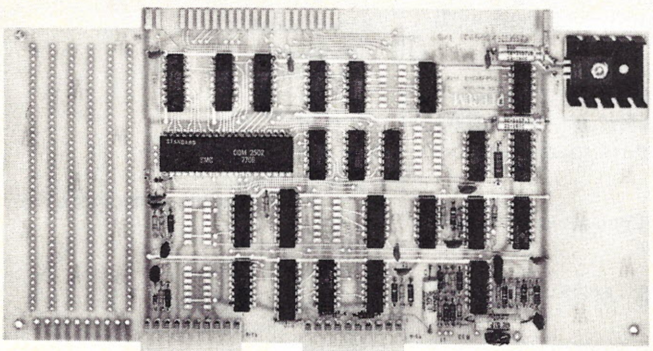
BASIC	Processing Time (milliseconds)					
	Logarithm	*Exponential*	*Multiplication*	*L+E+M*	*Power*	*Square Root*
Mits	19	28	4	51	55	46
Reg. North Star	99	73	5	177	167	92
FP North Star	9	8	2	19	18	4

Table 4. Comparison of logarithm/exponential function processing times with power function processing time.

Rod Hallen
Road Runner Ranch
Box 73
Tombstone AZ 85638

Kansas City Standard— At 1200 Baud

Although he finds some deficiencies in the CI-812 software and manual, the author is generally satistfied with Percom's I/O cassette board.



The Percom CI-812 cassette and RS-232 interface board. Strapping for the RS-232 data rate is in the center; straps to select the port address are in the lower center (A1 to A7); and the connectors for the recorder and the RS-232 device are at the top. The vacant area on the right can be used for modifications or additional features.

When I purchased the Percom CI-812 cassette and RS-232 interface board, it was my intention to write a short review of my initial experiences with it. However, in order to fit the board to my requirements, I first had to write some software. Some of the difficulties I faced and some of the solutions that I came up with should be of interest to anyone contemplating a cassette read-write interface.

I wanted to be able to read and write Kansas City Standard tapes at 300 and 1200 baud or perhaps faster. The Percom CI-812 ad struck my eye and I ordered the assembled board with remote control and test cassette (Table 1). My order arrived promptly; along with the board came a 40-page manual and a signal-level kit.

Hardware

The CI-812 is well designed and manufactured. I have had absolutely no problems with it and no component failures so far. It is made for the 8080 S-100 bus computer.

Although I purchased the assembled version, the assembly instructions were included, and I doubt anyone would have any difficulty putting it together. IC sockets are not provided or recommended by Percom unless purchased from them. The only construction required on the assembled board is the installation of the signal-level kit (a resistor and an LED) and the control and audio cables (not supplied) that go to the cassette recorder.

In addition to the cassette I/O that reads and writes tapes at 300, 600, 1200 or 2400 baud, an RS-232 compatible interface for a video terminal is provided (see "Who's Afraid of RS-232?" by Greg Pickles, *Kilobaud*, May 1977, p. 50). A simple adapter allows this to drive a 20 mil Teletype loop, also. The data rate of this interface can be strapped to 110, 300, 600, 1200, 2400, 4800, or 9600 baud. Although I haven't put the RS-232 interface to use yet, it is destined to bring my Selectric I/O to life.

Here are some of the features of the board that I particularly like:

1. Provision is made for an external monitor speaker to be connected to the board. It is nice to be able to hear what is going on when looking for trouble or trying to locate the beginning of a desired data file on the tape.
2. The signal-level kit is a joy! Start the tape, increase the volume control until the LED flickers, and you've set the playback level correctly.
3. The cassette record data rate can be set by an external switch, which I found most convenient.
4. A large area of the board is left over and can be used for modifications or additional ports as the need arises.

Factory wired, the CI-812 utilizes ports 0 and 1, but can be restrapped to use any two adjacent ports within the 8080's 256 possibilities.

I bought the remote control but found it to be of limited use. It will turn one or two recorders on or off under software control; however, this causes loss of manual control of the recorder unless I remove the remote control plug, and then I'm likely to forget to plug it back in again. Timing loops are required to allow the tape to get up to speed before reading or writing can take place. The manual quite frankly acknowledges the limitations in normal use.

Modification information is provided in case you want to interface a parallel keyboard, 110 baud Teletype or 134 baud Selectric.

My testing has shown that the average audio cassette is reliable at 300, 600 and 1200 baud, but that it takes a high-quality audio or digital tape to handle 2400 baud. A 300 baud tape can be loaded into memory and then dumped at 1200 baud to make for faster loading in the future. Keeping the original tape as insurance against accidental erasure or damage to the duplicate is a good idea.

The Manual and Test Cassette

The manual has sections on assembly, parts, schematics, modifications, theory and operation. It was written for the computer with front-panel switches and no resident operating system. The software provided consists of some test patterns at 300 and 1200 baud, a

checksum loader, and a complete micro operating system. If you have an operating system in PROM, you'd do just as well to pass up the test cassette.

The operating system has the following features: examine memory, change memory, load from tape, verify tape, dump tape, and execute program. Here is how it works: Enter the bootstrap loader via the front-panel switches; this loads the checksum loader from the tape, and this, in turn, loads the operating system. The routines (Programs A, B and C) included with this article can be entered from your keyboard if your computer already has an operating system.

Difficulties

I expected to read the manual once or twice, plug in the board, enter a little software and start reading and writing tapes. It didn't quite work out that way! As I've mentioned, the software in the manual and on the test cassette does not take into account the computer that already has an operating system. In addition, it is designed to be located at C000 to C1FF, which is where my monitor PROM is sitting. So I decided to write my own.

For those who use the software provided with the CI-812, here is one fault that needs correction: There are two different ways to set the output-data rate for the cassette interface. It can be the same rate as that selected by an option strap for the RS-232 interface, or it can be selected separately with an external switch. Which of these two means is actually used is controlled by software.

If the RS-232 device is a video terminal, it would normally be set too fast to clock the cassette I/O; having a switch-selectable cassette data rate is more convenient, anyway. This point is well explained in the manual, but the software still uses the RS-232 rate for the cassette I/O. Once you figure out the problem, this is easily fixed.

Lines C905 and C907 in Program A select the cassette mode and the data rate that is set by the cassette data-rate switch. If I output 02 instead of 03, then the cassette data rate is the same as that strapped for the RS-232 interface.

Software

Fortunately, enough information is given in the manual to enable you to write your own dump and load tape routines,

Address	OP Code	Assembly Language	Remarks
C9ØØ	21 ØØ ØØ	LXI ØØ ØØ	Start dump address
C9Ø3	DB Ø1	IN 1	Clear UART
C9Ø5	3E Ø3	MVI A 3	Select cassette mode
C9Ø7	D3 ØØ	OUT Ø	& set data rate
C9Ø9	DB ØØ	IN Ø	UART ready ?
C9ØB	E6 8Ø	ANI 8Ø	
C9ØD	CA Ø9 C9	JZ C9Ø9	If not loop
C91Ø	3E CB	MVI A CB	Output "CB"
C912	D3 Ø1	OUT 1	(Block start)
C914	DB ØØ	IN Ø	UART ready?
C916	E6 8Ø	ANI 8Ø	
C918	CA 14 C9	JZ C914	If not loop
C91B	7C	MOV A H	Output high order
C91C	D3 Ø1	OUT 1	address
C91E	DB ØØ	IN Ø	UART ready?
C92Ø	E6 8Ø	ANI 8Ø	
C922	CA 1E C9	JZ C9 1E	If not loop
C925	7D	MOV A L	Output low order
C926	D3 Ø1	OUT 1	address
C928	DB ØØ	IN Ø	UART ready?
C92A	E6 8Ø	ANI 8Ø	
C92C	CA 28 C9	JZ C928	If not loop
C92F	7E	MOV A M	Output data byte
C93Ø	D3 Ø1	OUT 1	pointed by H & L
C932	FE DD	CPI DD	Is it "DD"? (Block end)
C934	CA Ø3 CØ	JZ CØØ3	If so exit
C937	23	INX H	Increment address
C938	C3 28 C9	JMP C928	Get another byte

Program A. A memory-to-cassette tape dump routine. Addresses C901 and C902 will determine the location where the dump starts. If you want to start dumping at address 1A09 (hex), then enter line C900 as 21 09 1A (low-order address first). First the characters CB are placed on the tape; the start dump address is output; and then the data contained in memory is dumped until the characters DD (end block indicator) are output, at which time the routine stops and an exit to the resident operating system is made. DD should be placed in memory at the end of any program or block of data that you intend to tape.

CI-812 kit	$99.95
CI-812 assembled	129.95
IC socket kit	14.95
Remote-control kit	14.95
Test cassette	4.95

Percom Data Company
PO Box 40598
Garland TX 75042

Table. 1.

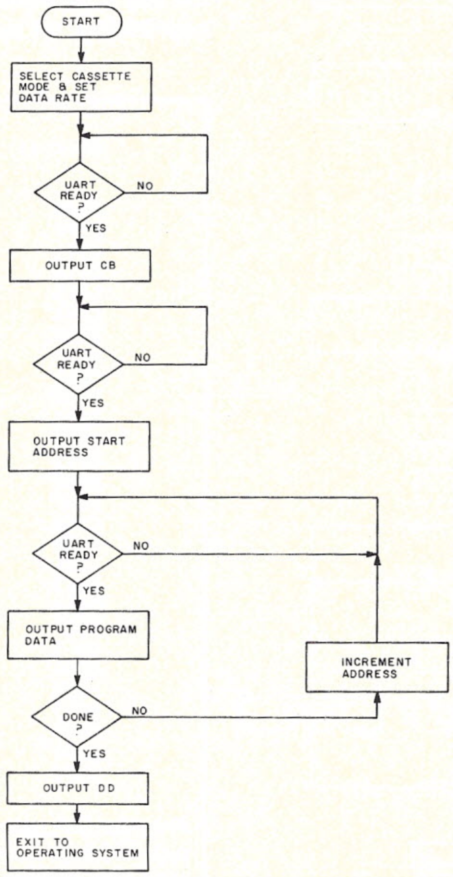


Fig. 1. Flowchart of the tape dump routine. Each symbol in the chart represents a block of assembly-language code. Compare the flowchart with the remarks in Program A to get a clearer picture.

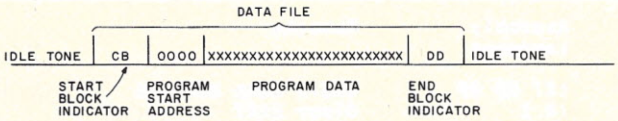


Fig. 2. A representation of how the start block indicator, program start address, program data and end block indicator appear on the tape. Before CB and after DD, the interface places a 2400 Hz idle tone on the tape. Ten to fifteen seconds of idle tone between programs on the tape makes it easier to separate them by ear with the monitor speaker.

as I did. Fig. 1 is the flowchart and Program A is the listing for my tape dump routine. The remarks column of the listing should be self-explanatory; the following information should also help.

These routines can be relocated, starting on any page boundary of memory, by replacing C9 each time it appears with the desired page number. For instance, to load these routines starting at 4000, replace C9 with 40.

Line C900 tells the routine where to find the first byte of the program that you wish to record on tape. Lines C905 and C907 select the cassette mode and set the data rate determined by the cassette data-rate switch. When the UART is ready, lines C910 and C912 record the hex characters CB on the tape. I call this my "start block indicator."

Next, C914 to C926 record the beginning address (in this case 0000) on the tape. C928 to C938 output the desired program data.

Each of my programs in memory ends with the hex characters DD. I call this my "end block indicator." When line C932 loads DD on the tape, the dump routine exits to my operating system, which gives control back to the keyboard. The data file on tape looks like Fig. 2. I chose CB and DD as my start block indicator and end block indicator, respectively, because they are not 8080-implemented operation codes and would not appear in the body of an assembly-language program.

Fig. 3 is the flowchart and Program B is the listing for my tape-load routine. C940 and C942 select the cassette mode and set the cassette data rate. (In this case, the rate is clocked by the data on the tape.) C94D, C94F and C951 keep looping until a CB is located; C95B and C95D load the high-order address byte into the H register, and C965 and C967 load the low-order address into the L register.

C96F to C978 enter the taped file into the memory pointed to by the H and L registers until DD is encountered. When this happens C974 exits to my operating system. You can write the exit in Program A and this program to any location that suits your purpose.

Fig. 4 is the flowchart of Program C, a tape-load routine intended to load tapes that do not contain CB or a load address. It normally starts with the first nonzero character but can be modified as indicated in the program caption. After being loaded with this program, a file can be dumped with Program A and then loaded with Program B from then on.

Implementation

In order to dump a program, the desired data rate is switch selected, the starting address of the program to be dumped is entered at C901 and C902, and the recorder started. After it has run for a few seconds, the command Execute (or run) C900 will record the program on tape. When the exit to the operating system is made (mine puts the cursor back on the video screen

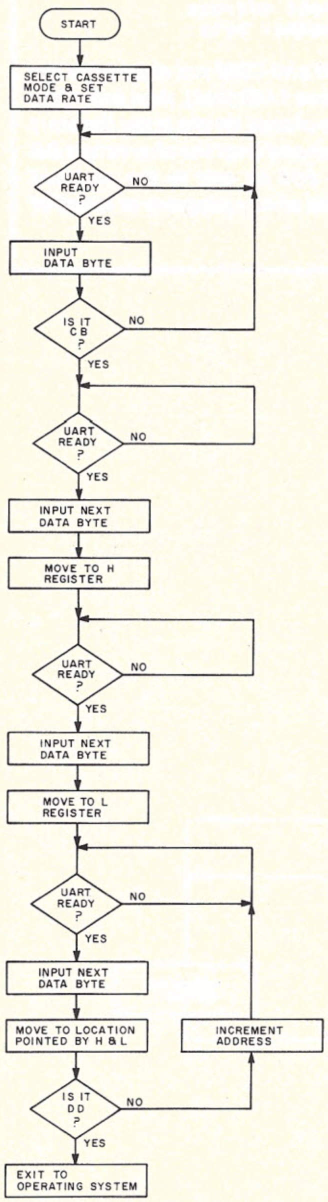


Fig. 3. Flowchart of the tape load routine listed in Program B.

```
Address   OP Code     Assembly      Remarks
                      Language

C94Ø      3E Ø1       MVI A Ø1         Select cassette mode
C942      D3 ØØ       OUT Ø            & set data rate
C944      DB Ø1       IN 1          Clear UART
C946      DB ØØ       IN Ø             UART ready?
C948      E6 4Ø       ANI 4Ø
C94A      CA 46 C9    JZ C946       If not loop
C94D      DB Ø1       IN 1          INput byte
C94F      FE CB       CPI CB        Is it CB? (start block)
C951      C2 46 C9    JNZ C946      If not loop
C954      DB ØØ       IN Ø             UART ready?
C956      E6 4Ø       ANI 4Ø
C958      CA 54 C9    JZ C954       If not loop
C95B      DB Ø1       IN 1          Input high order address
C95D      67          MOV H A       Move to H register
C95E      DB ØØ       IN Ø             UART ready?
C96Ø      E6 4Ø       ANI 4Ø
C962      CA 5E C9    JZ C95E       If not loop
C965      DB Ø1       IN 1          Input low order address
C967      6F          MOV L A       Move to L register
C968      DB ØØ       IN Ø             UART ready?
C96A      E6 4Ø       ANI 4Ø
C96C      CA 68 C9    JZ C968       If not loop
C96F      DB Ø1       IN 1          Input data byte
C971      77          MOV M A       Move to memory
C972      FE DD       CPI DD        Is it DD? (end block)
C974      CA Ø3 CØ    JZ CØ Ø3      If so exit
C977      23          INX H         Increment address
C978      C3 68 C9    JMP C968      Get another byte
```

Program B. A tape-to-memory loading routine. This routine will read the tape until it finds the characters CB (start block indicator); it will place the next two bytes into the H and L registers; then load data starting at the address pointed to by H and L. It will continue to load until it reads the characters DD (end block indicator), at which time it stops reading and exits to the resident operating system.

Address	OP Code	Assembly Language	Remarks
C98Ø	21 ØØ ØØ	LXI H ØØØØ	Start load address
C983	3E Ø1	MVI A Ø1	Select cassette mode
C985	D3 ØØ	OUT Ø	& set data rate
C987	DB Ø1	IN 1	Clear UART
C989	DB ØØ	IN Ø	UART ready?
C98B	E6 4Ø	ANI 4Ø	
C98D	CA 89 C9	JZ C989	If not loop
C99Ø	DB Ø1	IN 1	Input byte
C992	FE ØØ	CPI ØØ	Is it ØØ?
C994	CA 89 C9	JZ C989	If so loop
C997	77	MOV M A	Move to memory
C998	DB ØØ	IN Ø	UART ready?
C99A	E6 4Ø	ANI 4Ø	
C99C	CA 98 C9	JZ C998	If not loop
C99F	23	INX H	Increment address
C9AØ	DB Ø1	IN 1	Input data byte
C9A2	77	MOV M A	Move to memory
C9A3	C3 98 C9	JMP C998	Get another byte

Program C. A cassette-to-memory loader intended to load tapes that do not contain CB (start block indicator) or load address. This routine will read the tape until it finds the first nonzero byte and then start loading at the address indicated in line C980. If you have a listing of the program and know the first byte to be loaded, enter that byte at C993 and change C994 to C2 (JNZ). Now the routine will read until it finds the indicated byte, and then start loading.

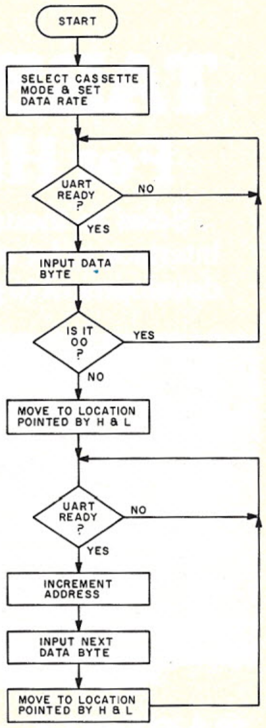


Fig. 4. Flowchart of Program C. Since the data file does not contain an end block indicator, no exit is provided for. The program is run until you hear the end of the file in the monitor speaker; then the computer is reset to end the program execution.

as an indicator), stop the tape.

When a tape is loaded with Program C, an automatic exit will not be made at the end of a file, and it is necessary to monitor the tape and push the reset button when the file has been loaded. Don't forget to write down the tape counter reading on the cassette recorder so you can find any particular file easily.

To load from tape to memory, set the data-rate switch to the rate that was used in recording, find the program on the tape if there is more than one, start the tape five to ten seconds before the beginning of the file, and command Execute C940. The routine will load all data on the tape between CB and DD into the memory locations from which it originally came.

Instead of using CB as a start block indicator for all files on a single tape, each file could have a distinctive start block indicator that would allow the load routine to find the desired file by itself. However, this is slow; it is much easier and faster to move the tape to the beginning of the file before executing the load routine.

Conclusion

Not too much has been said in this article about the RS-232 interface; not too much is said about it in the manual either. With a borrowed modem, I determined that it works properly and that I will have to develop some software in order to use it with my Selectric. I plan to pass along this information in a subsequent article.

Despite what I feel are shortcomings in the manual and software provided, I still think that the CI-812 is a good buy. It has been dependable, and, with the routines just discussed, it has given me what I was looking for when I first went shopping for a cassette interface board. ■

Bruce Turrie
Riverside Towers
Esmorietiaan 3#68
B2050 Antwerpen, Belgium

Swords and Sorcery!

This game is not for the faint of heart! However, if you make it through all the obstacles you'll be solvent and possibly have a date for the weekend.

Game playing is not my major purpose for owning a personal computer; eventually I want my machine to refrib frooIaps for me (it's almost impossible to find a competent gnome these days). Nevertheless, until I have learned the capabilities of my SWTP 6800, I will stick to writing game programs—one of the best ways I know to learn a new system. It is also fun, and it gives me a chance to exercise my imagination.

Some of the aspects of my system I wanted to investigate were: (1) How limited am I with no hard copy? (2) How does BASIC compare with the languages I've used (APL, FORTRAN, COBOL)? (3) How much is 12K, 16K, etc.? (4) What bugs are in SWTP 8K BASIC version 2.0? (more about this later). (5) How reliable is audio cassette recording (moderately), and how badly do I need a disk? (very).

For the most part, I was pleasantly surprised with the capabilities of my equipment; 20K of memory is larger than I expected. This program fits nicely in 16K. BASIC is also not nearly as bad as I feared. Given adequate preparation, you can compensate for the lack of hard copy.

Writing the Program

I do not claim that this is a well-written program. In fact, it is a good example of how programs can get out of hand during revision. The code has become difficult to follow in places because there are too many GOTO statements, which happened in spite of my good intentions and orderly approach.

I began by writing several pages describing the program. My initial idea included one large loop that would present the player with a series of decisions and events as the game progressed. The events would be handled in subroutines. Next I made a short list of the characters and events to incorporate as subroutines. These would either be called or not called by the main program based on random numbers. Then I drew flow diagrams based upon the above descriptions. At this point I was doing well.

Finally, I started writing code, and this is where I first began to go wrong. I wrote and tested in blocks of code about 50 lines long, and funny things started to happen at the interfaces between blocks. This was exaggerated when I began incorporating new ideas into the program. (If this happens when you are organized, imagine what happens when you're not.) Anyway, I am happy with the result. The program fits 16K and should be suited to the addition of new ideas using subroutine calls from the main program loop. These were my major objectives (aside from the experience obtained by writing the program). Therefore, I consider the program successful—it's even fun to play!

The Scenario

You are a gallant but broke hero attempting to rescue an elfin princess from the Necromancer's dungeon located within a dark forest. In order to accomplish the rescue, you must pass safely through the forest to the dungeon, remove the princess and escort her to safety. You must also pick up enough gold along the way to pay your creditors.

On your adventures you will meet various creatures—

benign and hostile and indifferent. How effectively you deal with them will determine the final outcome.

Characters

The creatures and people you might meet are shown in Table 1, which places the characters in the order of increasing power (to help or hinder you). One unlisted character who should at least be mentioned is the elfin princess, who actually plays a very minor role in the game. You will have to work out your own fantasies about her.

The Program

There are no remark statements in this program since there was not sufficient memory. However, it should be possible to determine the purpose of any given segment of code by reading the associated PRINT statements. Table 2 provides a list of significant line numbers to assist you.

A word of caution: Variables using the letter E cannot immediately follow a statement number. Apparently the BASIC line editor considers them end statements. In any event, such statements are never entered properly, and an error message is not always issued. Since I discovered this after I was committed to E1 for the game's enchanted sword, I kludged my way around the problem. E1 is only set in lines with more than one statement—such as line 925.

I also had trouble when I tried to put two IF-THEN statements on a line; the second did not work properly.

Playing Hints

1. Dryads are easily offended and have powerful curses. They will tell you which is the correct path only if they are certain.
2. The oracle is not always reliable.
3. Casting lots only works occasionally; in an unlucky game, it's better to choose the opposite of the lot cast.
4. An enchanted sword is extremely valuable against trolls; if you lose it, much of your original fighting ability is diminished.

A Final Word

If anyone has a working system for refribing froolaps, please write (I use only dextrorotatory froolaps but I could adapt a levortatory program).■

Helpful

1. Escaped slaves: Mostly just there.
2. The oracle: May tell you the correct entrance to use.
3. Dryad or Nymph: Very lucky, and good at choosing paths.

Hostile

1. Rats: Revolting, but not dangerous if you run.
2. Snakes: Not deadly, but their bite makes you unable to travel for one day.
3. Trolls: Most are moderately dangerous. Those trained to guard or fight are much worse.
4. Goblins: Will either enslave you or sell you to satyrs if you cannot pay them ransom.
5. Satyrs: Very bad if you have females with you. You will never see them otherwise.
6. Necrómancer: You do not want to meet this character.

Table 1.

Line Numbers	Purpose
30 - 120	Set up for Dryad.
220 - 388	Oracle.
520	Start of main program loop.
520 - 655	Dryad suggests path.
660 - 730	Cast lots to choose path.
910 - 940	Find enchanted sword.
940 - 960	Snake.
970	Check for troll.
1200	Check for rats.
1300	Check for pit.
1350	Check for necromancer.
1445 - 1450	Check for satyrs.
1475 - 1490	Gold.
1500	Check for rest.
1510 - 1530	Slave girl.
1540	Check for new fork.
1560 - 1616	Dead end.
1700	End of loop.
3000 - 3025	Yes or No response handler.
3100 - 3120	Dryad is offended.
3490 - 3710	Rest subroutine.
3715 - 3891	Goblins subroutine.
4000 - 4310	Rescue and end of game.
4500 - 4630	Troll.
4700 - 4730	Run.
5000 - 5004	Time killer.
5100 - 5340	Satyr subroutine.
6100 - 6232	Pit subroutine.
6300 - 6395	Necromancer.
7000 - 7020	Rats.
7990 - 8405	Battle with "trained" troll.

Table 2.

Program listing.

```
0001 D1=0:F9=0:D2=0:D3=0
0002 X1=0:V1=0:F1=0:C1=0:C2=0
0003 D4=0:F5=0:E1=0:F8=0
0004 G1=0:I2=0:P3=0
0008 PRINT "ENTER A NUMBER"
0009 INPUT I3
0010 PRINT "SWORDS AND SORCERY FOR THE 6800"
0011 PRINT "V 1.0 BY B.D. TURRIE"
0020 PRINT "GOOD LUCK. YOU'LL NEED IT!"
0021 PRINT
0022 PRINT
0023 GOSUB 5000
0025 DIGITS= 1
0030 IF RND(I3)>.7THEN200
0040 PRINT "A DRYAD WISHES TO GUIDE YOU."
0050 PRINT "WILL YOU LET HER?"
0060 GOSUB 3000
0070 ON I1GOTO80,120
0080 F1=0
0090 IF RND>.2THENGOSUB3100
0100 GOTO 200
0120 F1=1
0200 W1=F1
0210 PRINT "THERE ARE THREE PATHS INTO THE"
0211 PRINT "FOREST. ONE IS CORRECT."
0220 C1=INT(3*RND+.9999)
0222 Y1=INT(3*RND+.9999)
0225 IF RND>.4 THENY1=C1
0230 PRINT "DO YOU WISH TO CONSULT AN ORACLE?"
0240 GOSUB 3000
0250 ON I1GOTO400,260
0260 PRINT "HOW MANY MAIDENS SHOULD"
0270 PRINT "BE SACRIFICED?"
0280 INPUT V1
0285 V1=INT(V1)
0290 IF V1>2*RNDTHEN320
0295 PRINT "THE ORACLE IS OFFENDED"
0300 C2=C2-1
0320 IF V1*F1<10*RNDTHEN350
0330 GOSUB 3100
0350 IF C2<0THEN388
0360 IF RND<.4-(V1/10)THEN388
0370 PRINT "THE ORACLE SAYS PATH";Y1
0380 P1=1
0381 GOTO 400
0388 PRINT "THE SIGNS ARE UNCLEAR"
0400 PRINT "CHOOSE PATH 1,2 OR 3"
0410 INPUT X1
0415 X1=INT(X1)
0420 IF X1>3THEN400
0425 IF X1<1THEN400
0426 L1=100*(RND+1)
```

```
0427 IF X1=C1THEN444
0431 C2=C2-2:L1=L1+50
0444 C2=C2+1
0445 T1=0
0450 C2=C2+.5*RND+F1+P1+.2*V1*SGN(5.1-V1)
0490 C3=0
0520 PRINT "YOU HAVE COME TO A FORK IN THE  PATH"
0525 C1=INT(2*RND+.9999)
0530 Z1=W1-F1:S1=8-Z1-G1/(1+Z1)
0535 IF S1<4THENS1=4
0540 IF 3.5*RND>C2THENL1=L1+1
0545 L2=7+20*RND
0550 L0=L2
0560 IF F1=0THEN660
0570 PRINT "DO YOU WANT TO ASK THE NYMPH    WHICH ONE?"
0580 GOSUB 3000
0590 ON I1GOTO660,600
0600 IF RND>.5+C2/50THEN650
0610 PRINT "SHE SAYS",C1
0620 GOTO 740
0650 PRINT "SHE DOESN'T KNOW"
0655 GOTO 740
0660 PRINT "WILL YOU CAST LOTS TO DECIDE?"
0670 GOSUB 3000
0674 ON I1GOTO740,680
0680 PRINT "THE LOT SAYS"
0685 IF RND>.5+C2/10THEN700
0690 X1=C1
0695 GOTO 730
0700 IF C1=2THENX1=1
0710 IF C1=1THENX1=2
0730 PRINT X1
0740 PRINT "DO YOU CHOOSE 1 OR 2"
0750 INPUT X1
0755 X1=INT(X1+.0001)
0760 IF X1>2THEN740
0770 IF X1<1THEN740
0780 D0=0
0790 IF X1<>C1THEN820
0800 C2=C2+RND*SGN(.5-RND)
0810 C3=0
0815 GOTO 840
0820 C2=C2-.2
0830 C3=-1
0840 IF T1=0THEN890
0890 L2=L2-S1
0895 I2=I2+1
0900 IF E1<>0THENGOTO940
0910 IF RND>.15+.2*F1THENGOTO940
0920 PRINT "YOU FIND AN ENCHANTED SWORD"
0925 C2=C2+.2:E1=1
0940 IF RND-.5*(F1+C3)<.95THEN960
0950 PRINT "SNAKE!!!"
0955 GOSUB 3490
0960 T1=T1+1
0970 IF RND>.8+(C2+C3)/30THENGOSUB4500
1200 IF RND>.95THENGOSUB7000
1300 IF RND<.03THENGOSUB6100
1310 P3=0
1350 IF RND>.99+(C2-I2/10)/100THENGOSUB6300
1445 IF W1=0THEN1460
1450 IF RND>.95+(C2+C3)/50THENGOSUB5100
1460 IF RND<.95THEN1500
1470 PRINT "GOLD! HOW MANY SACKS"
1475 PRINT "WILL YOU TAKE"
1480 INPUT X1
1490 G1=G1+INT(X1)
1500 IF T1>10-2*F1THENGOSUB3500
1510 IF RND>.05THEN1540
1520 PRINT "YOU MEET AN ESCAPED SLAVE GIRL"
1530 W1=W1+1
1540 IF L2>0THEN840
1550 L1=L1-L0*.85
1560 IF L1<20THENGOSUB4000
1570 IF C3<>-1THEN520
1580 IF RND<.9THEN520
1590 IF RND>.5GOSUB6100
1595 PRINT "DEAD END"
1600 L2=L2+L0
1615 L1=L1+L0
1616 C3=0
1700 GOTO 840
2980 GOTO 840
```

```
2990 END
3000 PRINT "ENTER 0 NO OR 1 YES"
3003 INPUT I1
3010 IF I1<0THEN3000
3015 IF I1>1THEN3000
3020 I1=I1+1
3025 RETURN
3100 PRINT "THE NYMPH IS VERY OFFENDED"
3105 PRINT "SHE CURSES YOU AND GOES"
3110 F1=0
3112 W1=0
3115 C2<C2-1
3120 RETURN
3490 PRINT "YOU ARE HURT"
3500 PRINT "YOU MUST REST"
3502 PRINT "YOU'VE GONE";I2*S1;"KM."
3510 T1=0
3515 GOSUB 5000
3520 D3=D3+1+W1-F1
3530 IF D3<4-W1+F1THEN3700
3540 IF D4<>0THENGOTO3650
3550 PRINT "YOU ARE OUT OF FOOD"
3560 D4=1:C2=C2-RND
3570 GOTO 3700
3650 F5=F5+1
3660 C2=C2-RND+.2*F1
3700 IF RND>1-(I2+F8)/100THEN3715
3702 PRINT "TIME TO MOVE ON."
3705 IF F5<7THENRETURN
3710 PRINT "YOU STARVE TO DEATH":GOTO2990
3715 PRINT YOU ARE CAPTURED BY GOBLINS"
3720 IF E1<>1THEN3800
3725 PRINT "THEY WANT YOUR SWORD"
3730 PRINT "WILL YOU TRADE IT FOR FREEDOM?"
3735 GOSUB 3000
3740 ON I1GOTO3800,3750
3750 PRINT "AGREED":E1=-.8
3752 IF P3=1THENPRINT"THEY THROW YOU BACK IN THE PIT"
3755 GOTO 3705
3800 D9=INT(3*RND+.9999)
3805 IF G1<D9THEN3850
3810 PRINT "THE GOBLIN LORD FREES YOU"
3820 PRINT "FOR";D9;"BAGS OF GOLD"
3825 G1=G1-D9
3830 GOTO 3752
3850 IF W1>0GOTO3880
3860 PRINT "YOU ARE ENSLAVED"
3870 GOTO 2990
3880 PRINT "YOU ARE SOLD TO THE SATYRS"
3890 GOSUB 5130
3891 GOTO 3752
4000 IF R1<>0THEN4200
4020 PRINT "YOU ARE AT THE ENTRANCE"
4021 PRINT "TO THE DUNGEON"
4022 PRINT "THERE IS A GUARD"
4025 GOSUB 7990
4110 R1=1:W1=W1+1
4120 PRINT "YOU HAVE RELEASED THE PRINCESS"
4124 GOSUB 3500
4200 IF L1>0THENRETURN
4205 IF R1<>1THEN4275
4207 PRINT "YOU MADE IT!!!!"
4210 PRINT "YOU ARE A HERO"
4215 IF G1<2THEN4240
4220 PRINT "YOU MARRY THE PRINCESS"
4225 GOTO 2990
4240 PRINT "TOO BAD YOU ARE TOO POOR"
4245 PRINT "TO MARRY THE PRINCESS"
4247 PRINT "YOU LIVE THE REST OF YOUR LIFE"
4250 PRINT "IN QUIET POVERTY"
4255 GOTO 2990
4275 IF R1<>-2GOTO4305
4280 PRINT "YOU LET THE PRINCESS BE"
4285 PRINT "TAKEN ALIVE BY SATYRS"
4290 PRINT "YOU ARE JUSTIFIABLY TORTURED "
4295 PRINT "TO DEATH"
4300 GOTO 2990
4305 PRINT "YOU HAVE FAILED"
4310 GOTO 4247
4500 PRINT "TROLL!!!"
4510 PRINT "WILL YOU FIGHT?"
4520 GOSUB 3000
4530 ON I1GOTO4540,4560
```

```
4540 GOSUB 4700
4550 RETURN
4560 IF E1=0THEN4580
4570 IF RND>.2THEN4580
4574 PRINT "THE TROLL RUNS "
4576 RETURN
4580 IF RND+.3*E1>.5THEN4610
4585 IF RND>.3THEN4590
4586 GOSUB 3490
4587 RETURN
4590 PRINT "YOU ARE KILLED"
4600 GOTO 2990
4610 PRINT "THE TROLL IS DEAD"
4620 IF RND+.05*E1<.4THENGOSUB3490
4630 RETURN
4700 PRINT "RUN!!"
4710 L1=L1+4*S1*(RND-.7)
4720 T1=T1+1
4725 IF RND>.7GOSUB6100
4730 RETURN
5000 FOR I1=1TO200
5001 X1=RND
5003 NEXT I1
5004 RETURN
5100 PRINT "OH NO! SATYRS."
5105 IF E1<>1THEN5130
5110 IF RND<.5THEN5130
5120 PRINT "THEY RUN FROM YOUR SWORD"
5125 RETURN
5130 PRINT "THEY WILL GO IF YOU FIGHT"
5135 PRINT "THEIR CHAMPION"
5140 GOSUB 3000
5145 ON I1GOTO5200,5160
5160 GOSUB 7990
5165 IF RND<.2THEN5200
5170 RETURN
5200 PRINT "THE SATYRS WANT THE FEMALES"
5205 W1=0:F1=0:IFR1=1THENR1=-2
5210 PRINT "WILL YOU ALLOW THIS"
5220 GOSUB 3000
5230 ON I1GOTO5240,5260
5240 PRINT "THEY KILL YOU AND TAKE THE WOMEN"
5250 GOTO 2990
5260 PRINT "THE WOMEN ARE TAKEN"
5270 IF RND<.03THEN5310
5280 PRINT "THEY CURSE YOU"
5290 C2=-5
5310 IF RND>.3THENRETURN
5320 PRINT "THE SATYRS KILL YOU ANYWAY"
5340 GOTO 2990
6100 PRINT "YOU HAVE FALLEN INTO A PIT."
6105 P3=1
6110 IF RND>.5GOSUB3490
6120 PRINT "YOU MUST ESCAPE. WILL YOU"
6130 PRINT "TRY TO CLIMB OUT 1, OR"
6140 PRINT "YELL FOR HELP 2?"
6150 INPUT X1:X1=INT(X1)
6152 IF X1<1THEN6120
6153 IF X1>2THEN6120
6154 P7=0
6155 ON X1GOTO6160,6200
6160 IF RND<.5-P7/10THEN6180
6162 PRINT "YOU FALL WHILE TRYING TO CLIMB"
6164 PRINT " ":PRINT" "
6165 IF RND<.2THENGOSUB3490
6167 P7=P7+1:IFP7<5THEN6160
6170 PRINT "TOO DEEP. YELL FOR HELP."
6172 GOTO 6200
6180 PRINT "YOU'RE OUT":RETURN
6200 FOR I1=1TO5
6201 PRINT "HELP!"
6202 NEXT I1
6203 GOSUB 5000
6204 IF RND<.3THEN6200
6205 PRINT "A ROPE HAS BEEN LOWERED."
6207 P3=0
6210 Y1=INT(3*RND+.9999)
6215 ON Y1GOTO6220,6225,6230
6220 GOSUB 4500:RETURN
6225 GOSUB 3715:RETURN
6230 PRINT "YOU ARE RESCUED BY A WOMAN"
6231 W1=W1+1
6232 RETURN
6300 PRINT "IT'S THE NECROMANCER...."
6305 IF RND>.6THEN6392
6310 IF F1<>1THEN6326
6320 PRINT "THE NYMPH GOES MAD."
6325 W1=W1-1:F1=0
6326 IF E1=1GOTO6375
6330 IF W1=0THEN6350
6335 W1=0
6340 PRINT "THE WOMEN ARE SENT TO HIS"
6345 PRINT "UNDEAD AS PLAYTHINGS"
6350 PRINT "YOU ARE TRANSFORMED INTO"
6351 PRINT "A GELDED WATER BUFFALO"
6352 PRINT "AND DRIVEN INTO A PIT."
6360 PRINT "BOY DID YOU LOSE!"
6370 GOTO 2990
6375 PRINT "HE TAKES YOUR SWORD":E1=-1.5
6380 PRINT "YOU ARE CAST INTO A PIT"
6385 GOSUB 6110
6390 RETURN
6392 GOSUB 4700
6395 RETURN
7000 PRINT "UGH! RATS, MILLIONS OF THEM."
7010 GOSUB 4700
7020 RETURN
7990 W3=2:H1=1:H2=1
7992 W2=1+.3*RND
7995 H3=2+SGN(E1)
8000 PRINT "IT'S A LARGE TROLL"
8040 PRINT "THE BATTLE BEGINS"
8050 PRINT :PRINT"YOU CIRCLE FOR POSITION"
8051 PRINT :FORI1=1TO20
8052 X1=RND
8053 NEXT I1
8055 W3=W3-.05:H1=H1-.05
8057 H2=H2-.05
8060 PRINT "HE ATTACKS!"
8070 X1=1+INT(2.9999*RND)
8080 ON X1GOTO8090,8100,8110
8090 PRINT "HE TAKES A WILD CUT"
8091 GOTO 8120
8100 PRINT "HE THRUSTS AT YOUR BODY"
8105 GOTO 8120
8110 PRINT "HE TRIES A HEAD CUT"
8120 IF RND>.5+.3*H2/W2THEN8350
8130 X1=1+INT(2.5*RND)
8140 ON X1GOTO8150,8160,8170
8150 PRINT "YOU STOP HIS BLOW ON YOUR SWORD"
8151 PRINT "AND BACK AWAY"
8155 GOTO 8050
8160 PRINT "YOU DUCK UNDER HIS SWORD"
8161 PRINT "TO ATTACK"
8165 GOTO 8175
8170 PRINT "YOU PARRY THEN ATTACK"
8175 FOR I1=1TOH3
8180 IF RND>.1THEN8190
8185 PRINT "YOU MISSED HIM!":GOTO8250
8190 X1=1+INT(2.999*RND)
8195 ON X1 GOTO 8200,8220,8230
8200 PRINT "YOU GOT HIS LEG"
8205 W2=W2-H2/5:W3=W3-H2/5
8210 GOTO 8250
8220 PRINT "YOU'VE SLASHED HIS ARM"
8225 W2=W2-H2/3:W3=W3-H2/5
8227 GOTO 8250
8230 PRINT "YOU SCORE TO HIS BODY"
8245 W2=W2-.1:W3=W3-.25
8250 NEXT I1
8251 IF W2<.1W2=.1
8255 IF W3>.05THEN8050
8260 PRINT "HE'S DOWN!!"
8265 PRINT "YOU'VE FINISHED HIM OFF"
8290 RETURN
8340 PRINT "YOU'RE HIT!"
8355 H1=H1-.2:H2=H2-.2
8357 IF H1<.05THEN8370
8360 PRINT "YOU STAGGER AWAY":GOTO8050
8370 PRINT "YOU'RE DOWN!!!"
8380 PRINT "HE CLOSES FOR THE KILL"
8390 IF RND>.1+E1/20THEN8400
8392 PRINT "YOU MAKE A LUCKY THRUST!!"
8395 PRINT "HE'S DEAD!!!!":RETURN
8400 PRINT "YOU ARE KILLED"
8405 GOTO 2990
```

See the Rex at these dealers

ARIZONA
Tempe . . . . Computerworld, Inc. (602) 944-6070

CALIFORNIA
Colton . . . . Electronic Register Systems (714) 825-4502
Escondido . . . . Business Enhancement Compuservice (714) 741-6335
Huntington Beach . . . . Cybernetics, Inc. (714) 848-1922
Lawndale . . . . Jade Computer Products (213) 679-3313
San Diego . . . . Byte Shop (714) 565-8008
San Francisco . . . . Computer Store of San Francisco (415) 431-0604
San Jose . . . . Byte Shop III (408) 377-4685
San Rafael . . . . Byte Shop—Computer Store (415) 457-9311
Sherman Oaks . . . . Peoples Computer Shop (213) 789-7514
Tarzana . . . . Tech-Mart (213) 344-0153

COLORADO
Englewood . . . . Byte Shop, Inc. (303) 761-6232

IOWA
Davenport . . . . Computer Store of Davenport (319) 386-3330

KANSAS/MISSOURI
Overland Park . . . . Personal Computer Center, Inc. (913) 649-5942

MARYLAND
Baltimore . . . . Computer Workshop of Baltimore, Inc. (301) 486-5350
Rockville . . . . Computer Workshop, Inc. (301) 468-0455

MICHIGAN
Kalamazoo . . . . Computer Shop of Kalamazoo (616) 375-5340
Royal Oak . . . . Computer Mart of Royal Oak (313) 576-0900

MINNESOTA
Minneapolis . . . . Computer Depot (612) 927-5601

MISSOURI
Parkville . . . . Computer Workshop of Kansas City, Inc. (816) 741-5055

NEBRASKA
Lincoln . . . . Microtech, Inc. (402) 423-8404
Omaha . . . . Omaha Computer Store (402) 592-3590

NEW YORK
New York . . . . Computer Mart of New York (212) 686-7923

OHIO
Fairborn . . . . Forbees Microsystems, Inc. (513) 879-1157
Lyons . . . . Affordable Computers (419) 923-2821

OREGON
Beaverton . . . . Byte Shops Northwest (503) 644-2686
Portland . . . . Byte Shops Northwest (503)223-3496

PENNSYLVANIA
Murraysville . . . . Computer Workshop of Pittsburgh, Inc. (412) 327-0455

TEXAS
Dallas . . . . Computer Shops, Inc. (214) 234-3412
Dallas . . . . Rondure Company (214) 630-4621
Houston . . . . Interactive Computers (713) 772-5257
San Antonio . . . . Computer Solutions (512) 828-1455

VIRGINIA
Springfield . . . . . Computer Workshop of Northern Virginia, Inc. (703) 321-9047

WASHINGTON
Bellevue . . . . Byte Shops Northwest (206) 746-0651
Bellevue . . . . The Microcomputer Center (206) 455-3710

WISCONSIN
Janesville . . . . Austin Computers (608) 752-6002

rcc™ realistic controls corporation

PET SCHEMATICS

Another First From "PET-SHACK".

For only $34.95 you get:

24" x 30" schematic of the CPU board, plus oversized schematics of the Video Monitor and Tape Recorder, plus complete Parts layout—all accurately and painstakingly drawn to the minutest detail.

PET ROM ROUTINES

Another Breakthrough From "PET-SHACK".

For only $19.95 you get:

Complete Assembly listings of all 7 ROMs, plus identified subroutine entry points; Video Monitor, Keyboard routine, Tape Record and Playback routine, Real Time Clock, etc.

To entice you we are also including our own Machine Language Monitor program for your PET using the keyboard and video display.

You can have the Monitor program on cassette for only $9.95 extra.

Send check or money order
TO: PET-SHACK Software House **P37**
Marketing and Research Co.
P. O. Box 966
Mishawaka, IN 46544

A NEW SOUND **A NEW MUSIC**

BITS.AND.BYTES*(BACH,BINARY + BOOLEAN)

12 INCH STEREO RECORD **25 MINUTES PLAYING TIME**

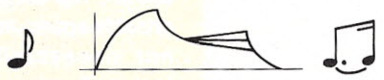
BACH **TRUE SCALES** **JOPLIN**

JOHN PRATT 3-VOICE COMPUTER-CONTROLLED MUSIC SYNTHESIZER
25 SCALES, 25 PARAMETERS PER NOTE PER VOICE
INSTANTANEOUS VOICE, TEMPO AND SCALE CHANGES
HARDWARE/SOFTWARE ABSTRACT ON BACK OF RECORD COVER

Features one of the more difficult of the Bach Three-Part Inventions for the piano played in true scales, also played on vibs & 3 guitars. "Entry of the Gladiators" is used to demonstrate accelerando and rallentando with many voice changes.

Music consists of the above two full-length selections; with the remaining 11 numbers being shorter selections using various voicings and tempos, which were created to debug subroutines and as exploratory music to determine how to best use our software & hardware.

The present-day even-tempered scale and 24 true scales are defined by software, although not all of the true scales are used in this record.



THE AVANT-GARDE MUSIC OF THE FUTURE
DO IT IN A COMPUTER, TO MUSIC IN NATURAL SCALES

dealers desired.

$6.00 postpaid (ck/M.O.)
(Calif res include 6% sales tax)

SCHERTZ COMPUTER MUSIC
P.O. BOX 8743 (NPS)
MONTEREY, CALIFORNIA 93940
S56

μ COMPUTER BOARDS

NEW! **CPU WITH SERIAL PORT**
8080A * * * S-100
SINGLE BOARD

Now it's easy — with a CPU Board which includes an on-board serial port. This 2 MHz CPU Board talks directly to your terminal by 20 ma current loop or RS-232. Baud rate selectable from 110 to 9600.

ASSEMBLED AND TESTED . . . ONLY – $195

NEW! **AUDIO CASSETTE INTERFACE**
WITH 3 PARALLEL I/O PORTS
S-100 * * * SINGLE BOARD

Your best choice for mass storage. This board includes 3 parallel 8 bit ports, a tape motor control (on-off) and a driver for external data I/O monitor lamp. The 3 PIO port common handshake signal lines are independent of data lines.

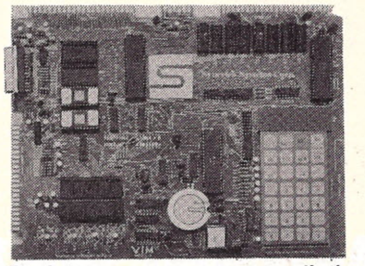
ASSEMBLED AND TESTED . . . ONLY – $195

OUR 20th YEAR OF ELECTRONIC EXCELLENCE

the Nucleus® INC. **N13**
461 Laboratory Road
Oak Ridge, TN 37830
615-482-4041

VIM-1 $269

KIM-1 COMPATIBLE
UP TO 4K RAM, 24K ROM



- **Ready to use as soon as you open the box**
- **KIM-1 Compatibility (KIM-1 is a product of Commodore)**
- **From Synertek** • **6502 Based system**
- **Single 5 volt power supply**
- **Supermon 4k monitor program**
- **1K bytes of 2114 static RAM on sockets**
- **Sockets for up to 4K RAM and 24K ROM**
- **15 bidirectional I/O lines on application port**

This is the VIM-1. NCE/CompuMart now offers the most versatile single board computer available today. Based on the popular and reliable 6502 CPU, the VIM-1 is designed to permit flexible solutions to a wide range of application problems. A system monitor (SUPERMON) is stored in 4K bytes of ROM furnished with the VIM-1 so you're free to concentrate on the application itself. But should you require customized system software, sockets are provided on the board for three ROM or EPROM packages that can expand total ROM to 24K bytes. And by changing jumpers, the VIM/1 can be set up to respond to your own system software as soon as the power is turned on.

For working with data and programs, VIM-1 comes equipped with 1K of RAM and sockets are available on the board for plug-in expansion of up to 4K. An expansion port is provided which will allow additional ROM, PROM, RAM or I/O to be attached to the system.

Features include a 28 key double function, audible keypad with up to 24 special functions, a bright 6-digit hex LED display, and 135 or 2400 baud cassette interfaces. RS-232 and current loop interfaces are present as well as a connection for the VIM-2 Keyboard module with TV interface which will be available later and eliminate the need for a separate terminal. The most unusual feature of the VIM-1 is an interface which enables you to output alphanumeric characters and custom graphics to an oscilloscope. The characters could be English, Greek, Cuneform or could even be stick-men, cars, dog houses, or laser guns.

And that's not all. A total of 51 active I/O lines (expandable to 71 with the addition of a plug-in 6522 chip) permit an almost endless variety of other peripheral devices to interface to the VIM-1, from floppy disk drives to full-ASCII keyboards and other computer systems.

The VIM-1 Reference Manual is complete with schematics, listings of SUPERMON and details concerning compatibility of the VIM-1 and Commodore KIM-1 hardware and software. A resident Assembler/Editor and 8K BASIC in ROM are on the way. From this you can see that the VIM-1 earns its initials as the Versatile Interface Module. Order now from NCE/CompuMart for early delivery.

10 day return privilege. Phone orders, and bank cards accepted. Foreign orders invited. Also purchase orders from established companies. Immediate Delivery. Our toll-free credit card order line is 1-800-521-1534. Mail your order to address below.

Just say charge it! Which card?

Member Computer Dealers Association

NCE/Compumart N7
1250 N. Main St., P.O. Box 8610
Dept. KB88 Ann Arbor, Mi. 48107
(313) 994-4445

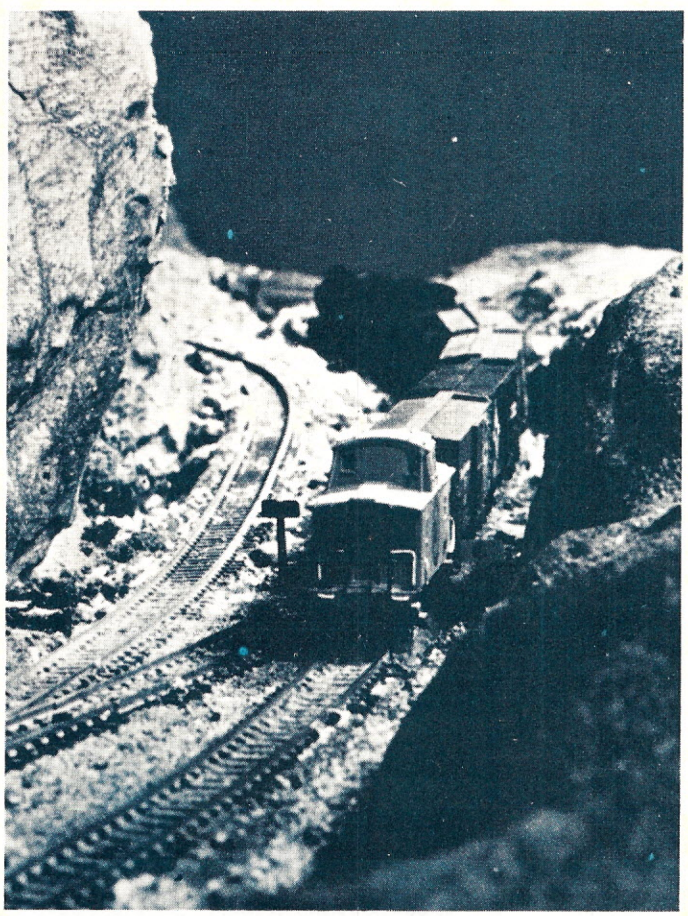
Dana W. Zimmerli
2020 Robinson St.
Redondo Beach CA 90278

Two Hobbies: Model Railroading and Computing

It really is a fascinating match-up; and and for those of us who haven't considered combining the two hobbies, perhaps this series of articles will provide the inspiration.

In the first part of this article, model-railroad background information and a top-down system design were presented.

In order to implement this design, we must look for circuitry to perform the required functions. Fortunately, certain techniques that have evolved over a period of time are applicable to computer control. In this article I will provide some suggestions about how to build the circuits. In the schematics shown, the component values are only approximations and do not necessarily represent tested circuitry.



Track Occupancy Detection

As signaling evolved for model railroads, track occupancy became important. Several techniques have been used, but the three currently most common are described below; in addition, one "blue sky" scheme is discussed. In the circuits shown, a light bulb is sometimes shown to represent the occupancy indication. In reality, this indication may also be a gate input signal, an opto-isolator or a relay.

The first circuit for track detection was the Twin-T circuit. It was first described by Linn Westcott in the June, July and August 1958 issues of *Model Railroader* magazine. This circuit was derived from several predecessors that used other components to detect the same parameter, namely, a current path between the rails of the track. Fig. 10 shows this cir-

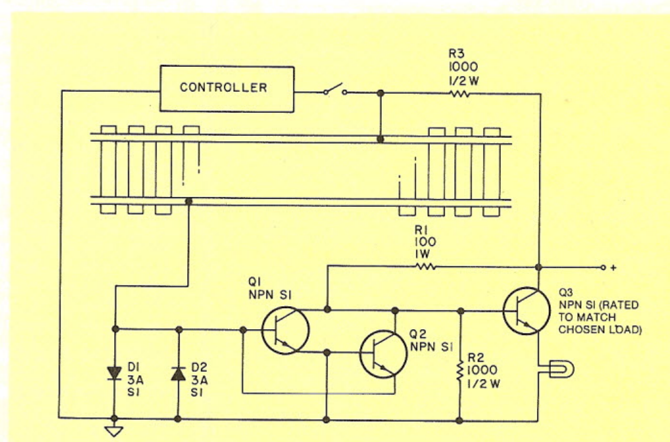


Fig. 10. Twin-T track detection.

cuit with a separate power supply for the detection circuits. Use of a separate supply in this way allows the detection of a standing train, as well as a moving, powered-up train.

The principle of operation is based on the two transistors, Q1 and Q2, which detect the current-induced voltage across D1 and D2. The current flowing in the track circuit is used to turn on one of the two transistors. If the current flows through D1, the base of Q1 is forward biased, causing Q1 to turn on. If the voltage polarity is reversed to change the direction of the train's movement, current will flow through D2, pulling the emitter of Q2 negative, which will turn on Q2. When either of these transistors turns on, the emitter-follower stage will go to zero, causing the lamp to turn off.

To drive a logical circuit from this form of track detection, the circuit shown in Fig. 11 will provide a positive voltage when the track is occupied. Several other variants on this basic circuit are possible. Careful examination of the Q1-Q2 transistor pair reveals that it can be made up using a Darlington pair to decrease the size and circuit package count. Another modification is to use an operational amplifier or a comparator instead of Q3. Also, if a high-frequency lighting power is already being used on the track, the extra resistor, R3, may be eliminated.

The biggest drawback of the Twin-T circuit is that in order to detect something on the track, it must conduct electricity. Most cars are not sold with current paths like this. Suggested modifications to the rolling stock are: lighting to passenger cars; track wipers connected to a hidden resistor; or resistive paint on, and between, the wheels. None of these solutions are satisfactory, so other means of detection are being used.

Another popular form of detection is the track contact. Various forms of contact are shown in Fig. 12. In the first case, the shim stock contact is forced away from the rail by a plastic wheel when a car passes through this section of track. This method, which can be made sensitive enough to work with the lightest cars, is intolerant of dirt and is fragile.

The second case is to cut out a short section of rail and insulate it from the other rail. Now if a metal wheel crosses the rail section, it will provide a path to the short section. This method requires metal wheels; in small scales this short section will be difficult to hold in place. The third case is the most reliable because the magnetic reed switches are totally enclosed. The only requirement is to mount magnets on some or all of the rolling stock to trigger the reed switches.

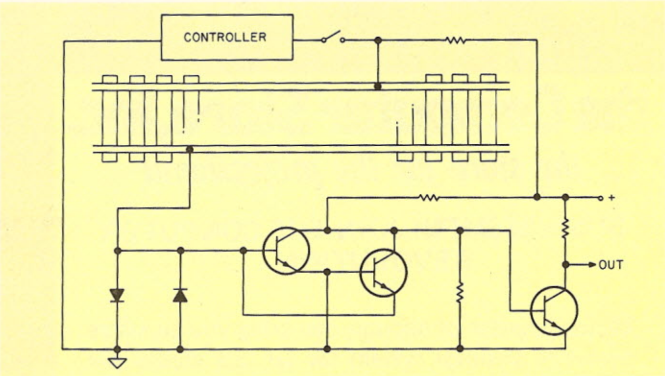


Fig. 11. Twin-T with logic level outputs.

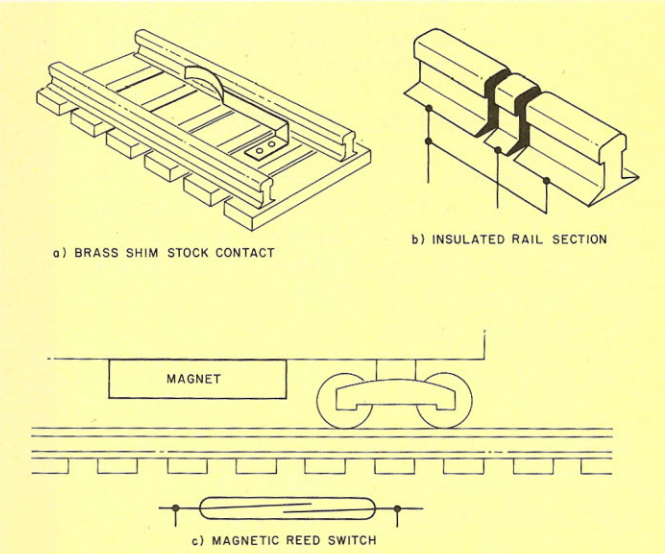


Fig. 12. Track contacts.

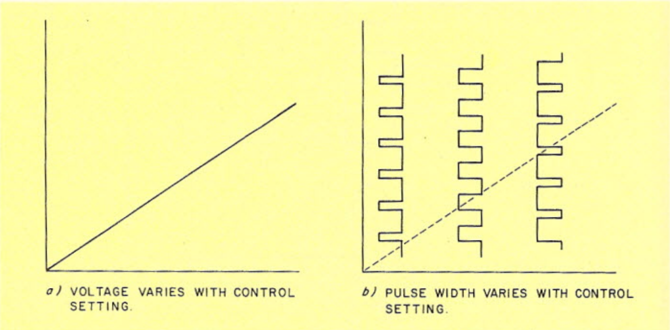


Fig. 13. Control of effective voltage.

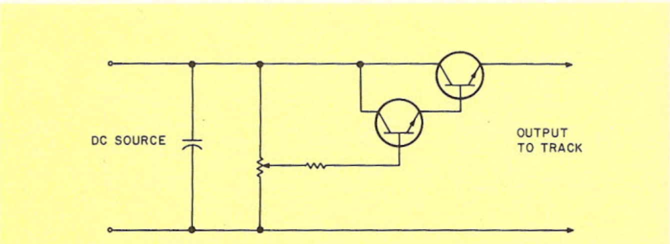


Fig. 14. Basic transistor control.

The third class of detection, optical detection, is similar to the track contacts of Fig. 12. However, since the optical detection uses a newer technology and does not impose any restrictions on the cars, it is treated separately. Various light-sensitive devices, ranging from the phototransistor to a cadmium-sulfide photocell, can be used. These light-detector circuits are identical to those used for any other light-detection requirement, so I will not repeat them here. The sensitivity of the detector depends on the orientation of the detector/light source, the intensity of the light source and the amount of reflected light that may also reach the detector.

Both of the last two forms of detection only sense the passage of cars through the sensing mechanism, so some form of memory is required to track the passage of a train through the track sections. Elaborate circuits (not necessarily complicated, just a lot of replication) have been designed to perform this tracking function. These circuits are called checkout circuits. In a computer-oriented system, the computer could be the memory device.

As with all volatile memory systems, checkout systems lose all knowledge of location when the power is turned off. Therefore, in order to initialize the system, either an initial condition must be defined, or

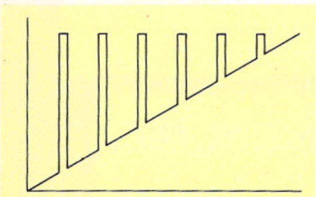


Fig. 15. Summation of voltage and pulse.

an initializing pass must be made over all the sensors.

The final form of detection is an untested scheme that may or may not work. While driving down the street one day, I noticed that traffic lights use sensors built around an inductive coil, I think this works on the same principle as a metal detector. If these could be built cheaply enough, it seems that it would satisfy almost all the requirements of track-occupancy detection. The only requirement on the cars is the presence of a sufficient amount of metal (not necessarily magnetic) to be detected. If a car does not have enough metal, it is probably too light, anyway, and should be weighted by adding some metal. If anyone wishes to investigate this, please let me know what you find out.

Power Control

In part 1 of this article, we

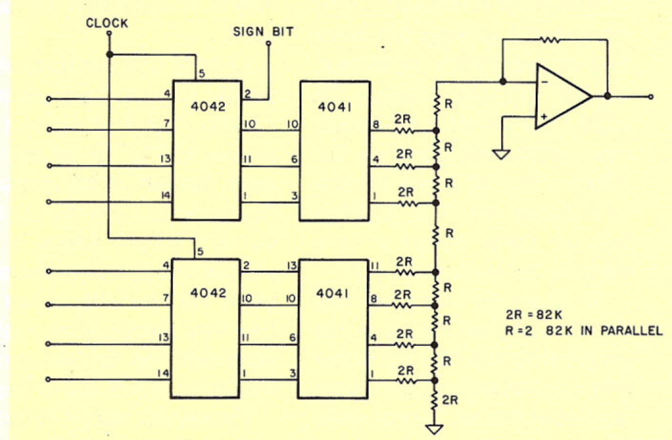


Fig. 16. Digital-analog conversion for speed control.

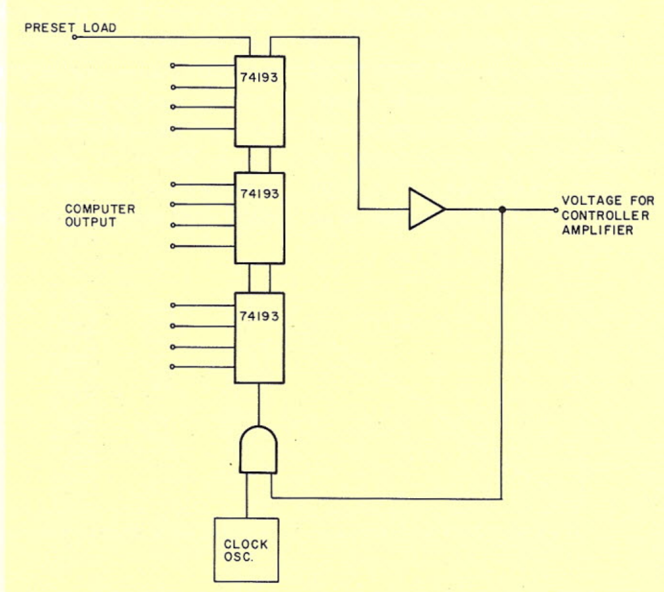


Fig. 17. Computer-controlled pulse-width power control.

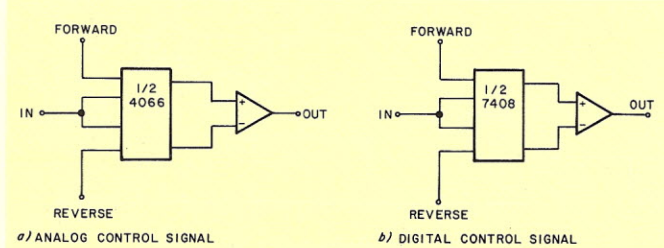


Fig. 19. Reversing with operational amplifiers.

discussed how to break up the layout into control blocks for controller connections. Now that we have gotten to the circuitry section, let's look at the way to vary the speed.

Since the standard form of motor is the dc motor, the method for varying the speed is to vary the effective voltage across the motor. Two basic methods of varying the effective voltage are: pulse-width variation and dc voltage-level variation. These methods are shown in Fig. 13. But that's not the whole story . . .

The first problem is that the current required is a function of load, so when the train is climbing a hill, it requires more current than when it is going downhill. This leads to the requirement for a constant-voltage transistorized controller.

This type of control, as shown in Fig. 14, is simply a means of providing a current-independent constant voltage level. The circuit is a simple voltage-regulator circuit. But wait a minute, that's not all . . . dc motors are not noted for starting or running slowly. In order to get them to do that, pulses have been used to overcome their inertia. Then there are momentum effects (remember, we said the computer could do that). A possible waveform for the basic voltage-variable controller is shown in Fig. 15. This waveform provides a good low-speed start and proceeds to full speed at full dc voltage.

At this point it should be apparent that the reason for the pulse-width controls is to provide a good low-speed response. These controls have been built in several forms, including SCR circuits. The usual objection is that the sharp rise-time of the pulses produces a power loss in the inductive reactance of the motor, causing the motor to overheat. The motor can also chatter quite a bit because of the pulses. For further power-control circuits, refer to the books referenced at the end of this article.

In order to computer-control the speed adjustment, the easiest concept is the use of a digital-analog converter as the source of the variable voltage applied to the transistors. This will require as many D/A converters as there are controllers, in addition to a set of latches to hold the digital value. A suitable simple version of a D/A converter is shown in Fig. 16. Precision components are not necessary for effective control, since motors are insensitive to small voltage differences.

Another possibility is to use the concept of the pulse-width controller. If the latching function was replaced by a preset countdown register, the output could become the OR of all the counter stages. If the oscillator could run at 1 kHz, and a number range was from 1 to 100, the computer would only have to set the count once every 100 msec in order to provide an effective range from full-off to full-on. This method would be much easier to implement in a microcomputer system. A block diagram of this approach is shown in Fig. 17. Remember, this system would only require an amplifier because the pulses are an intrinsic part of the circuit.

One further problem exists in providing full-power control—reversing the polarity of the track to reverse direction. In simple manual systems a DPDT switch is wired as shown in Fig. 18. For a fully computer-controlled system, some other way must be used. The most popular solution to the direction-control problem is the relay. However, I have another solution that is totally solid-state.

In most applications, the final current requirements are a maximum of one to three Amps, depending on the scale. If a source of high-power operational amplifiers were available at a reasonable cost, maybe they would be useful. Then the

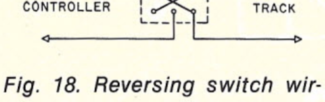


Fig. 18. Reversing switch wiring.

direction would be a matter of feeding the control voltage to either the inverting or the non-inverting input. Several ways of doing this are possible. Fig. 19 shows two, one based on an analog signal level and the other designed for the pulse-width control described above.

Now the only remaining question is the availability of a high-power operational amplifier. One company makes a one Amp operational amplifier, the LH0021. For small scales, particularly N-scale, this would be sufficient. However, there may be some question of availability, since only recently has this device shown up on the surplus markets. Fortunately, several well-known techniques to boost the output of an operational amplifier have been published. One of these is shown in Fig. 20. Of course, feedback would have to be provided as is normal in operational amplifier circuits. To increase the power even more, an additional Darlington stage could be added ahead of the transistors shown.

Closing the Switch With Authority

Since the track switch machines are basically a form of solenoid, certain problems sometimes exist when the machines are actuated. The main requirement for these switch machines is to provide an inrush of current for an instant—just long enough for the armature to slam into position. However, if current passes through the coil for very long, the coil will overheat and burn out. These problems are easily handled by a sudden current rush such as a capacitor discharging.

Many capacitive discharge switch machine supplies have been used. The only addition required for our computerized operation is a triggering SCR, which is used instead of a push button. The necessary circuit is shown in Fig. 21. The parts values shown should handle two or three switch machines, if the transistor is rated at 10 Amps and the SCR is one that will trigger on a logic-level signal.

Let's Build a Speed Trap

Previously I established the need for a speed-detection capability. Now that we have considered providing a combination computer/human operation, the computer is also interested in the speed of the trains. If the computer is running a train according to a timetable, it must have some feedback to tell how fast the train is moving. This requirement also means that the speed-detection circuit must be relatively inexpensive since a large number of them will probably be required. It is not sufficient to take a measurement at the beginning of the route and then depend on the train to maintain the same speed. Too many variables enter in for such a scheme to work reliably.

Probably the most accurate method of measuring speed is measuring the elapsed time between two predefined points. If these two points are defined by optical sensors, the circuitry required to measure time is quite simple (see Fig. 22). After the count is obtained, the microprocessor applies the basic speed-distance formula to obtain the speed. For this application, most of the formula becomes a constant that is divided by the measured elapsed time.

Computer Considerations

Now let's consider the way all these factors fit together. If we attempt to use this proposed system for a club layout, we will want to support from ten to maybe 100 track sections for five to 50 operators. One of the advantages of the scheme proposed here is the ability to have many more sections because the operators do not have to worry about where they are. This allows the operators to run trains much closer together without danger of collision or overrunning the next operator's block. Also, there are connections to be made for signals, speed detection, track occupancy and track-switch control. It is certainly conceivable that this system will need to sense and control over 250 different wires, each with several bits of data.

Now, how do we get that much data into and out of our computer? It is not likely that the I/O of an ordinary microcomputer would be ready to handle that many devices. In addition, interrupt processing would be impossible for that many lines, so some form of polling would be required (polling is the sequential process of cycling through all the I/O devices).

First, let's try to do something about distributing all the data. From experience, or at

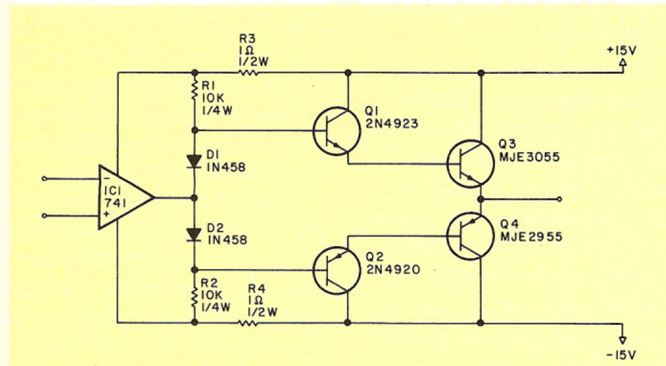


Fig. 20. High-power operational amplifier.

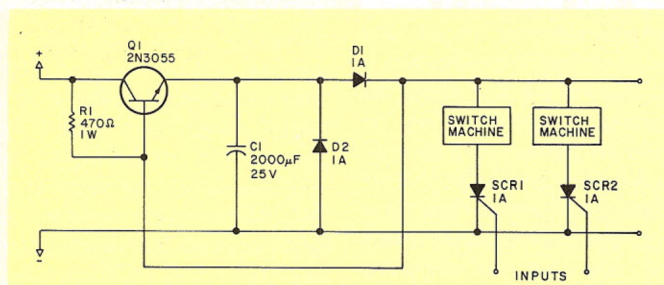


Fig. 21. Capacitive discharge switch machine power.

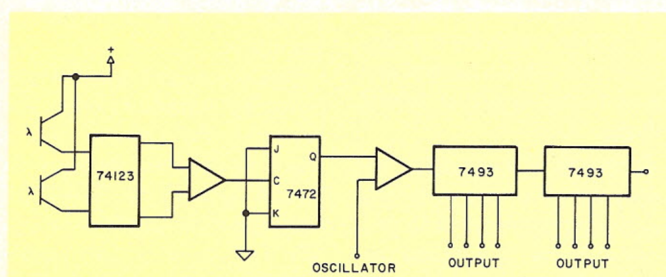


Fig. 22. Speed trap circuit.

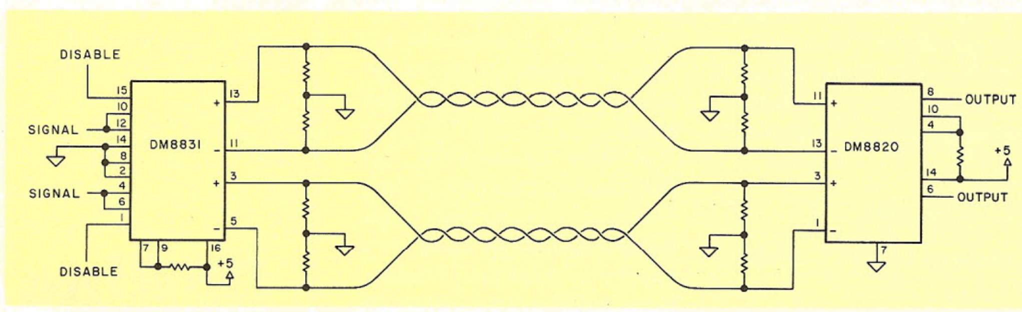


Fig. 23. Differential line drivers/receivers.

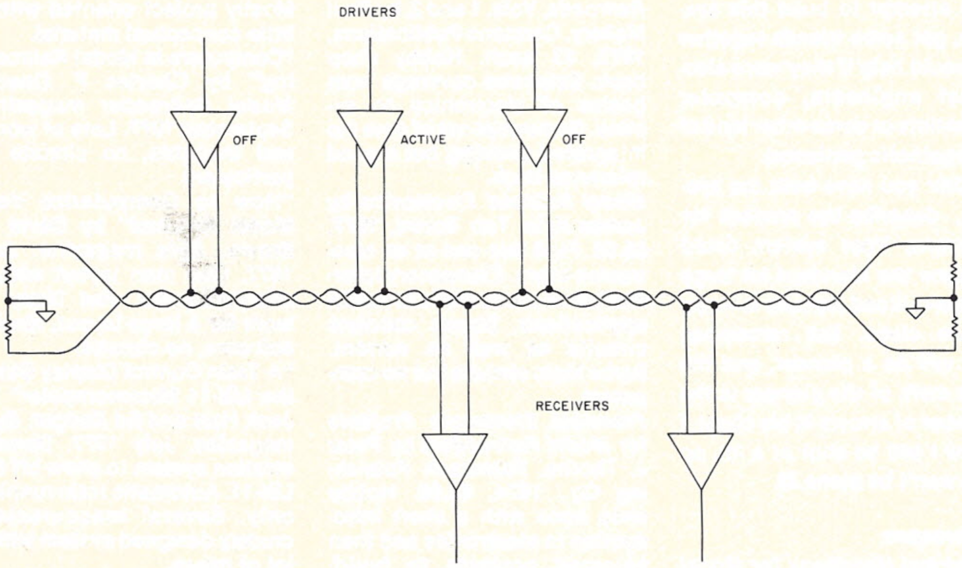


Fig. 24. A data bus for model railroads.

least from others' experiences, we all know about the problems of noise in digital signals. In the kind of system we are proposing, cables may have to run 200 to 300 feet. Maintaining an immunity to noise over that distance requires special line drivers and receivers. This is especially true since the model-railroad environment is noted for being extremely noisy due to the motors, switch machines, high-frequency lighting and sound effects. In particular, these requirements dictate differential line drivers and receivers, such as the DM8820 and DM8831.

These two devices work on a differential amplifier principle that allows all noise common to both wires to be cancelled out. They work best with shielded twisted-pair cables as shown in Fig. 23. Note that both the 8831 and the 8820 are dual devices. Also, since the cables must be treated as transmission lines, terminations have been shown for balancing the line impedances.

Since the 8831 is a Tri-state device, more than one driver can be attached to one cable (more than one receiver can also be used on the same cable because the input impedance is very high). In this configuration, the cable becomes a data bus (What! Allow a bus on a railroad layout?) with a termination on each end to prevent reflections. This is shown in Fig. 24.

The ability to put all the I/O for the layout onto three or four lines depends on the time-division multiplexing of the data onto one line (the additional lines are for timing signals). Another integrated circuit pair, the 74150 and the 74154, are designed to do this multiplexing and demultiplexing. To drive these devices, a counter is used to turn on and provide a control signal to these circuits. This counter is driven by one of the other cables.

On the computer end of the data bus, the computer can do the job of multiplexing and provide the timing signals for the counters. There may be some slight additional circuits needed, but, primarily, the data will be provided at the correct times by a software-polling loop.

Analog signals can also be encoded into variable width pulses to use the same drivers. Encoded analog data could then be detected by the receiver just like digital data, except that it would then have to be filtered and smoothed to recreate the analog signal. One requirement for analog data is communication between the dispatcher and the operators. This would probably be handled more efficiently by an ordinary telephone switchboard.

So what kind of a computer system will be necessary? The answer to this question depends on: (1) the set of capabilities that will be implemented; (2) the size of the layout to be supported (primarily the number of track sections, not the physical size); (3) the number of operators to be supported simultaneously; (4) other functions to be supported such as signaling, hump yards, communications and turntable indexing.

If the layout is a small one-man setup designed for two or three operators, a single small microcomputer will be sufficient. Even in this environment, the computer will have to poll several control positions, probably six to ten, and several controllers, probably six to 20. Signaling will add an additional 20 to 40 signals to be set. Fortunately, the timing requirements are low, and the changes are infrequent.

If we want to use the full-blown capabilities described above, some form of multiprocessor system may be necessary. A possible configuration to satisfy these requirements is shown in Fig. 25.

This configuration, which is similar to the systems used to control satellites, uses the 8008 as a multiplexer to send and receive data from the data buses. As data comes in, it is stored in the common memory (DMA is necessary to allow both processors access to the same memory).

Certain data values will also be marked as critical data. These data will be checked for changes each time they are read; if a change occurs, an interrupt will be sent to the other processor. This interrupt is the signal to the more powerful processor that something needs attention. If this results in a change to a controlled value, the new value is put into its place in the shared memory.

While this processing is going on, the 8008 continues to read the data lines. As it reads, it should also output data to the controllers, signals and other output devices. Therefore, as new values are stored in the memory, when the 8008 reaches them in the polling list, they will be sent out to the

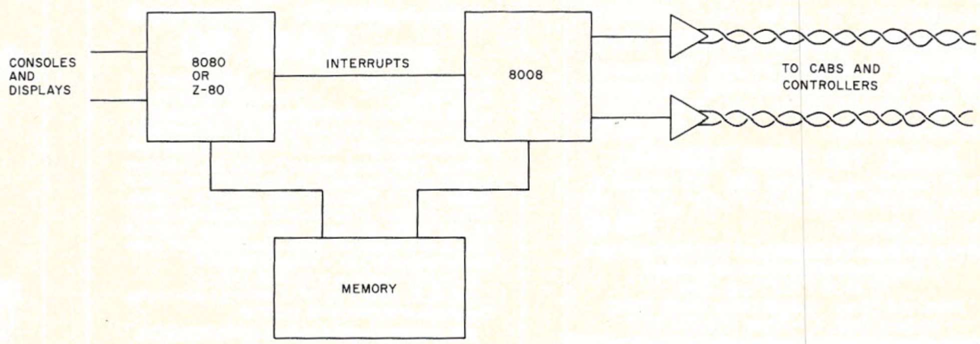


Fig. 25. Computer configuration for a large layout.

proper places. Requests for data and new information will be supplied directly to the larger microcomputer for inclusion in the operational computations.

The Future

Well, that's the background, the concepts, most of the circuits, and a possible configuration. For any specific case, it will take a great deal of development, several trade-off studies and an array of hardware. If you are still with me at this point, you are either interested, have nothing better to do or you read every word of this magazine, no matter what. If you attempt to build this system, get some friends together (it would help if they were electrical engineers, computer scientists or some other kind of sympathetic geniuses).

After you have built the system, describe the process for everyone. Your system could become the standard for large model-railroad layouts in many future clubs. But remember, you will be a pioneer, and the true mark of the pioneer is the number of arrows in his chest. I know I will be shot at a lot, so you won't be alone.■

References

Electrical Handbook for Model Railroads, Vols. 1 and 2, by Paul Mallery. Carstens Publications, 1973, $3 each. Hobby shop book. Extensive coverage from basics of electronics to exotica. Somewhat out of date (no integrated circuits), but a good reference book.

Model Railroad Electronics by James Kyle. Tab Books, 1977, $5.95. Find this one at an electronics store. Elementary information on electronics to model applications. Shows obsolete circuits as well as current. Some logic circuits but no computers.

Practical Electronic Projects for Model Railroaders by Peter J. Thorne. Kalmbach Publishing Co., 1974, $3.50. Hobby shop book with a short introduction to electronics and then selected projects to build. Mostly project oriented with a little conceptual material.

"Computers in Model Railroading" by Charles F. Douds. *Model Railroader* magazine, September 1977. Lots of words and thoughts, no circuits or particulars.

"How to Computerize Your Model Railroad" by David C. Brown. *Byte* magazine, July 1977. One man layout with a microcomputer. Not likely to work for a large layout. Mostly software, no circuits.

"A Train Control Display Using the LSI-11 Microcomputer" by Jack Hart and Ed Badger. *Byte* magazine, July 1977. Demonstration system to show off the LSI-11. Automatic train running only. Several inaccuracies, crudely designed system with a lot of relays.

Update: Lunar Lander

This short program will improve the landing speed of the May 1977 lunar-lander program.

Malcolm Shore
Defence EDP
Wellington New Zealand

In reply to Jim Huffman's article on the Lunar Lander program in MicroBASIC (*Kilobaud* No. 5, May 1977) I would like to point out one mistake and suggest some improvements.

The error is in line 520 — this test means that the moon module can land at any speed, and have a perfect landing!

Improvements: The landing speed should be any speed up to 5 ft/sec; the moon module has a certain amount of suspension! Also, velocity toward the moon would be positive, and velocity away from the moon negative. So here is the new improved Mk II Lunar Lander for you Mk II astronauts! ■

```
300  PRINT TAB(8);"LUNAR LANDER MKII"
310  PRINT TAB(8); "+++++++++++++++++"
320  PRINT
330  PRINT "WOULD YOU LIKE INSTRUCTIONS?"
340  PRINT "1= YES 0=NO"
350  INPUT A
360  IF A = 0 GO TO 430
370  PRINT "YOU HAVE 120 LBS OF FUEL"
380  PRINT "YOU ARE APPROACHING THE LUNAR"
390  PRINT "SURFACE AT 50 FT/SEC, AND"
400  PRINT "ARE CURRENTLY 500 FT FROM"
410  PRINT "THE SURFACE, TO CANCEL"
420  PRINT "GRAVITY BURN 5LB FUEL"
430  PRINT "HAPPY LANDINGS!"
440  LET F = 120
450  LET V = 50
460  LET D = 500
470  PRINT "FUEL"; F
480  PRINT "SPEED"; V
490  PRINT "DISTANCE"; D
500  PRINT "ENTER YOUR BURN".
510  INPUT B
520  IF B > F GO TO 500
530  LET F = F-B
540  LET C = B-5
550  LET D = D-V+C/2
560  LET V = V-C
570  IF D > 0 GO TO 470
580  IF V < 5 GO TO 620
590  PRINT "****!!CRASH!!****"
600  PRINT "YOU HIT THE MOON AT";V;"FT/SEC"
610  GO TO 640
620  PRINT "WELL DONE — YOU LANDED OK"
630  PRINT "LANDING SPEED";V;"FT/SEC"
640  PRINT "DO YOU WANT ANOTHER TRY?"
650  PRINT "1=YES, 0=NO"
660  INPUT A
670  IF A = 1 GO TO 430
680  END
```

Program listing.

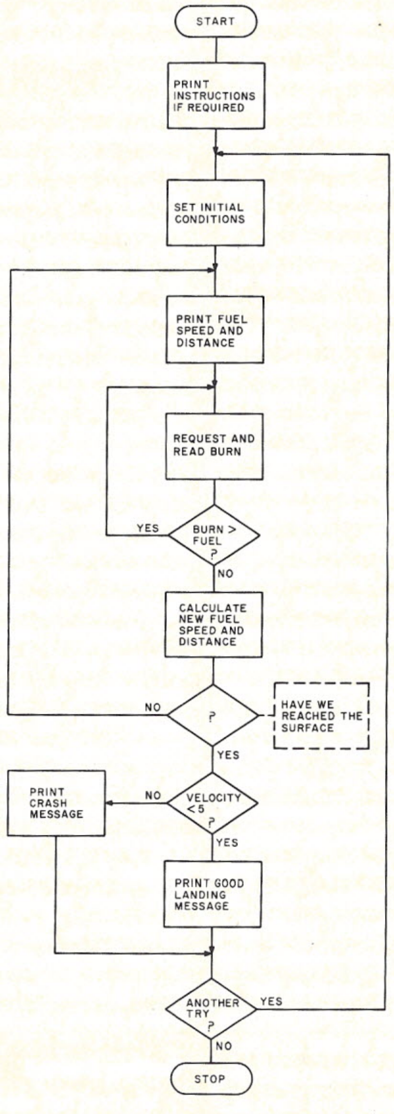


Fig. 1. Flowchart.

The Do-It-Yourself System: Heath's H8 Is a Winner!

If you're still looking around, you'll be very interested in the features and capabilities of the H8. Maybe it's the system in your future?

Robert M. Morgan
3449 Brunswick Rd.
Lexington KY 40503

The following account summarizes my experiences in acquiring a small computer system for home use. To a large degree, this article is modeled after Phil Hughes' review of his custom MP-68 system in the May 1977 *Kilobaud* ("Make Your Investment Count," p. 38). I found Phil's evaluation of a *total* system very helpful in organizing my own approach.

I had been in both staff and line management positions in data-processing software development for about six years when the concept of home computing first appeared feasible as a result of Mits' introduction of the Altair 8800. My associates' opinions about home computing varied. Some wanted a machine of their own, but most did not. Working at home with a 16K machine with a limited instruction set contrasts sharply with working with the office unit—a 16 meg, 30 nanosecond multiprocessor having virtual storage, billions of bytes of on-line random-access mass storage and an 18,000 line/minute multifont printer. Many also viewed programming at home as a busman's holiday—they like to leave their work at the office.

However, I identified with the few who wanted their own rig. The major computer vendor that supports my company unknowingly helped confirm my emotional need for a home machine by loaning me a portable computer to take home for a month to learn a new language (APL). Not only did I become an APL freak, I got hooked on having a machine at home!

Despite my emotional requirements for a system at home, it took a real-world need to justify an investment. I had entered into an agreement with a Los Angeles consulting firm to develop a FORTRAN-based graphics package to support one of their management-control systems. I was spending a lot of time designing, coding and testing algorithms that had little to do with moving a pen or drum around and making marks on paper. Most of the algorithms concerned data analysis and record relationships and chaining.

I thought (and have since proven) that I could be more productive by modeling these algorithms in an interactive BASIC environment rather than in a batch, $1500/CPU-hour environment that required inconvenient, time-consuming drives to a computer center. The last straw came when Sylvanhills Laboratory introduced reasonably priced plotter kits ($750)—I knew then that there was a plotter in my future.

Getting Started

My first act was one of great foresight: subscribing to *Kilobaud*. This was augmented by other readings, and an informal trip through BugBooks I and II supported by a homebrew digital lab. I also visited computer stores in Los Angeles, Atlanta, Louisville, Raleigh and New York. These investigations led to confusion—I saw a lot I didn't like, and little I did. Many systems just did not have what I call

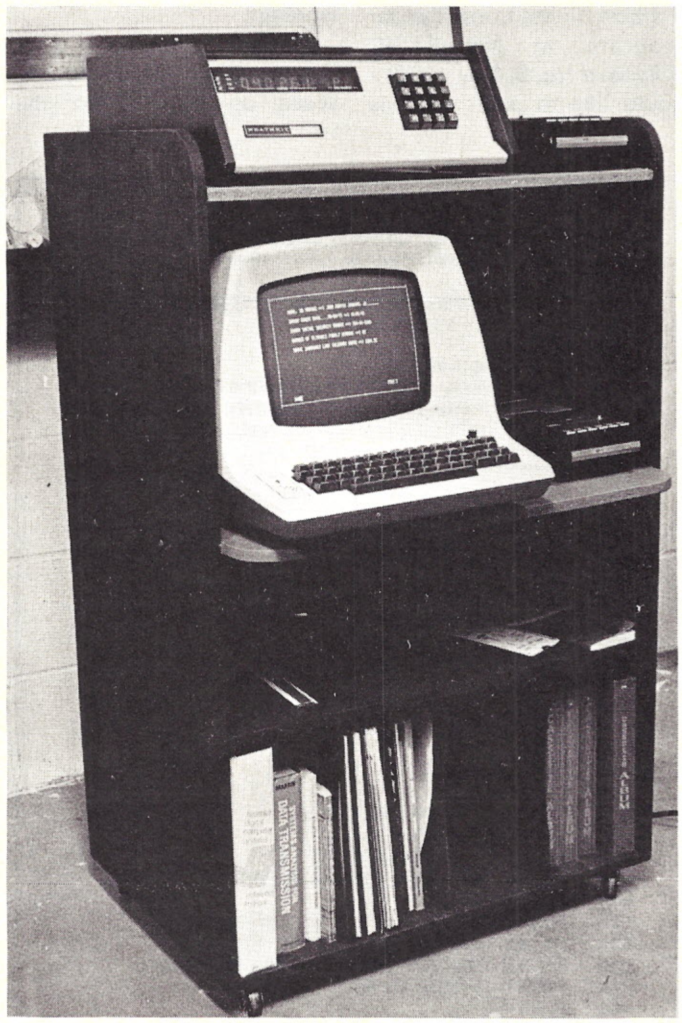


Photo 1. The complete system—H8 computer, ADM-3A terminal, two cassette recorders, custom-built console, tape library and reference books.

system continuity—a coherent approach to software and hardware. Those that had their act together were out of my budget ($1000 for CPU + memory + I/O, and $500 to $700 for terminal).

I soon realized that I had to go back to the basics and document my real requirements and constraints. These are summarized in Table 1. I concluded that I could afford, in the foreseeable future, an initial configuration; I couldn't afford an interim configuration (an undocumented 16-bit machine with DOS and floppy); and that the ultimate machine would not likely be available (i.e., affordable by individuals) until the mid-80s.

Enter the H8

About the time I reached my configuration conclusions, I received a brochure from Morrow's Micro-Stuff describing their 8080 microprocessor board. This, along with their cassette and serial interface board and memory boards would serve as a basis for a nice S-100 system—but they had no compatible software at that time (they have since announced several software packages).

Then Heath first advertised a picture of the H8 in the August *Kilobaud*. The following issue included a 14-page Heath catalog, and I was hooked. The Heath unit had everything I wanted. About a month later, I received a commission check for my graphics software, and I ordered a 12K H8 with Serial/Cassette and Extended Benton Harbor BASIC. At that point I felt that a terminal could wait, because I could at least get some experience in machine-language coding with the H8 front panel.

Photo 2. The programmer's work station: everything in reach!

Initial Configuration

Applications
- Model Larger System
- Drive Plotter

Languages
- BASIC
- Assembler

Hardware
- CPU ------------------ 8-bit
- Serial I/O-------------- For console support
- Cassette -------------- For program & data storage
- Parallel I/O------------- For plotter & printer

Monitor
- Resident ROM for 1 partition execution with interrupt & I/O management

Memory
- At least 32K of RAM

Final Configuration—The "Ultimate" Machine

Applications
- Broad range of scientific and commercial jobs, including multipartitioned operation with batch background, interactive foreground.

Languages
- FORTRAN
- BASIC
- Assembler
- PL/I
- APL

Hardware
- CPU ------------------ 32-bit
- I/O-------------------- various serial channels, plus byte & word parallel channels
- DISK------------------ Required if bubble memory not developed as disk replacement
- TAPE ----------------- IBM compatible 9-track, 1600BPI

Operating System
- Virtual multitask with variable partitions

Memory
- At least 512K of RAM

Table 1. Initial requirements analysis.

The H8 required about 30 hours to assemble. I did manage to foul up a few things during assembly, and was bailed out by a cooperative staff at Louisville's Heathkit Center. My construction errors cost me $54. I'd like to warn all H8 owners to take *special* care when inserting a card onto the motherboard. Should you get only one pin off, you will cause the -18 volt power line to be shorted to ground, and you'll know it by the smell of D-5 and D-6, and by the smoke . . . that little trick set me back $24. To complete my system, my wife gave me 4K for Christmas, and I just purchased a wired Lear-Siegler ADM-3A from a terminal broker.

The ADM-3A purchase was almost as difficult to make as the H8. I had an opportunity to purchase a Heath H9 CRT for $530 *assembled,* but, upon closer examination, I concluded that the graphics offered on the H9 were too limited to be of much general value, and that 12 lines were just not enough display to satisfy my needs . . . I also needed the terminal to communicate with two large mainframes (UNIVAC and IBM). In addition, I determined that Heath apparently has no plans to offer a kit to modify the H9 for uppercase/lowercase display. Thus, I selected ADM-3A, with its complete cursor control and uppercase/lowercase conversion capability.

While visiting the Heathkit center in Louisville, I determined that none of my cassette tape units would work at 1200 baud, and that the unit Heath recommends (their stock number ECP-3801) was required. However, I got mine at a local discount store, at a $20 savings (G.E. Model 3-5121). I purchased three of the Heath-recommended C-30 cassettes, but found that some inexpensive cassettes from the local Olson Electronics Store (Webcor C-30) performed much better (considerably fewer checksum errors), had screw-together cassette cases (handy for snipping off leaders), and cost much less (49 cents compared with $2/cassette).

While waiting for my terminal to arrive, I decided to avoid the tangle of wires seen in so many computer stores, and in some pictures of installations seen in *Kilobaud*. Knowing the dimensions of an ADM-3A, I constructed a package for the system that provides some flexibility for expansion (see

Major Components

Qty	Manufacturer	Model	Description	Cost
1	Heath	H8	Computer + Software	$ 375
2	Heath	H8-1	4K Memory Board	280
2	Heath	H8-3	4K Chip Sets	190
1	Heath	H8-5	Serial/Cassette Interface	110
1	General Electric	3-5121	Cassette Recorder	42
1	Lear Siegler	ADM-3A	Terminal	765
				$1762

Minor Components

Qty	Manufacturer	Model	Description	Cost
1	Heath	H8-13	Extended BASIC	$ 10
1	Heathkit Center	---	System Repairs	54
3	Heath	ECP-3802	Cassette tapes (C-30)	6
22	Webcor C-30		Cassette tapes	11
3	Radio Shack	44-612	12-Cassette Album	7
1	---	---	4' x 8' Panel for Cabinet	4
3	---	---	Sets duplex receptacles	5
1	Calectro	E2-161	Nonshorting Rotary Switch (baud rate selector)	2
---	---	---	Miscl. plugs/sockets, etc.	5
				$ 104

Total Initial System Cost	$1866

Table 2. System components.

Photo 1). Also, I can roll it into my den and back into my basement office with ease. The CRT is at a good operating height, the H8 at eye level, an extra shelf is available for floppy, and reference books are stored below. A flatbed plotter can be mounted *above* the H8 by attaching via screws to the back of the cabinet, and a shelf for a printer can be added to the side.

Software

I've not had the terminal long enough to explore most of the software in depth. It took me about a weekend to learn BASIC. The H8 operation manual (595-2014) and software reference manual (595-2048) combine to form an inch-thick reference bound in a Heath-supplied three-ring binder. The operation manual is hardware oriented, and includes test routines, troubleshooting, operating instructions, a nice section on theory of operation and system configurations. The appendix includes a discussion on "The Functions of a Computer," a detailed discussion on 8080 architecture, timing, etc., and the formal CPU specifications.

The software reference manual includes a general discussion of the various software packages, product installation notes, patches for nonstandard hardware configurations (i.e., ASR console support) and some useful reference tables. Following this general discussion are sections on the software included with the computer, and the one optional software product (Extended Benton Harbor BASIC). Software includes: PAM-8, panel monitor (in ROM); BUG-8, console debugger; TED-8, Heath Text Editor; HASL-8, Heath assembly language; Benton Harbor BASIC and Extended Benton Harbor BASIC.

I'm most familiar with Extended Benton Harbor BASIC. It requires about 78 seconds to load the interpreter from cassette at 1200 baud. I found the BUILD command useful in inputting new programs—BUILD 10,10 will cause an initial line number 10 to be displayed, any text entered will

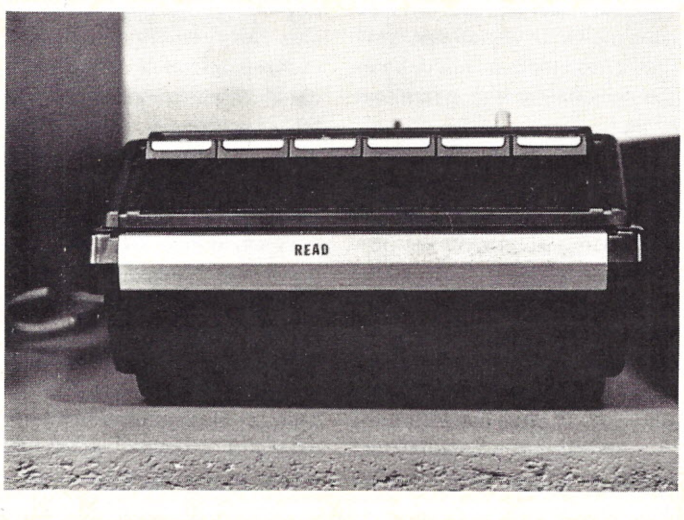


Photo 3. Even the cassette decks are labeled . . . to add a professional touch.

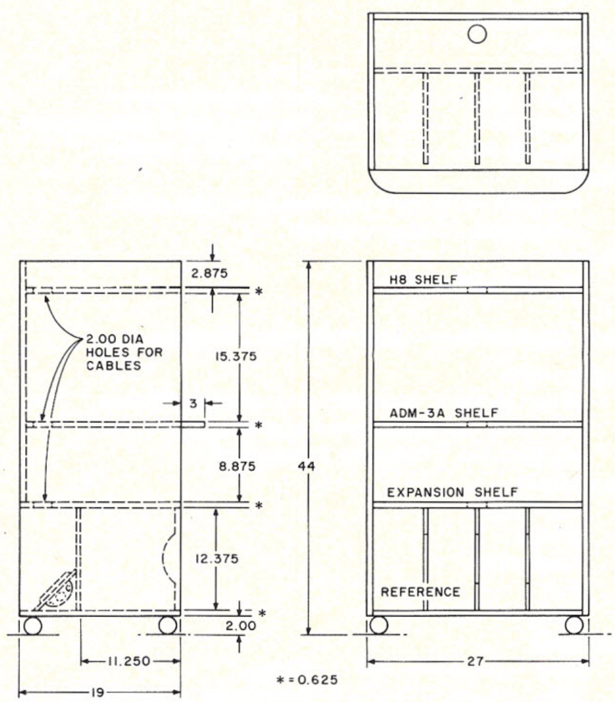


Fig. 1. Plan for system console. All dimensions in inches (scale). All material 5/8-inch particle board, except for shelf dividers, which are 1/8-inch hardboard. About 100 1¼-inch #7 wood screws and marine-grade plastic resin glue provide strength required to hold heavy equipment. Flush back and 2-inch cable holes provide for uncluttered look. Three duplex sockets provide power to system—located in bottom back cavity. Two-inch furniture casters will be replaced with heavy-duty 3-inch hard rubber casters. Finished in flat black and eggshell blue.

replace existing text for that line number, which will be incremented by the second number in the BUILD command upon carriage return.

Special commands allow data display from a BASIC program on the LED display on the H8, and the entering of data from the H8 keypad (values 0 to 9 from marked numbers, 10 from + key, 11 from - key, 12 from * key, 13 from / key, 14 from # key and 15 from . key). PAUSE will terminate execution until any character is entered on the console keyboard, or PAUSE iexp will cause a PAUSE of 2*iexp milliseconds.

There's a command to transfer output to a port other than the console output, and a command to transfer both the console output and input to another port. STEP allows you to step through one or a few lines of a BASIC program at one time. String operations are supported, but matrix functions are not.

A really nice feature is command completion. If, for example, you enter the command 150 PR, the H8 completes the spelling by completing the command: 150 PR*int*. The command completion also includes the required trailing blank.

BASIC lacks two features that I think would provide a quantum leap in utility: a RENUM command to renumber the program statement numbers and all their references, and a straightforward method of using the cassette deck for I/O from the BASIC program. I *think* I've discovered a way of using the USR function, but it will take some additional research and testing. If I'm successful and no other writer beats me to the punch, I'll share it in a future article.

Final Configuration

The major and minor components of the system are listed in Table 2. One of two modifications I've made to the hardware is to add a baud-rate switch on the serial I/O card for the console. I've run the ADM-3A at all speeds from 150 baud to 9600 baud. It is a little flaky on the output at 150 baud, and flaky on the input at 9600 baud, but it works! I normally use 4800 baud except when I want to slow down the output to watch the cursor closely . . . then I use 300 baud. The ADM-3A is configured as shown in Table 3.

INTERNAL SWITCHES				EXTERNAL SWITCHES			
LEFT		*RIGHT*		*UPPER*		*LOWER*	
SPACE	ADV	CUR CTL	OFF	BIT 8-0	1	AUTO NL	OFF
UC DISP	U/L DISP	LOCAL	OFF	PARITY	INH	RS232	CL
DISABLE	KB LOCK	103	OFF	STOP-1	2	HDX	FDX
DISABLE	CLR SCRN	202	OFF	DATA-7	8	. . . all baud switches	
50 HZ	60HZ	CODE	SEC	PAR-ODD	EVEN	off but one required	
12 LINE	24 LINE	EXT	OFF	LC EN	UC		
		EOT	OFF				

Table 3. ADM-3A configuration.

My other hardware modification was to add a DPDT switch to include or exclude the two jumpers on the serial/cassette board that allows for selection of one or two cassette decks. In one position, only one deck is used for recording and playback. In the alternate position, the jumpers are left open and two decks are assumed. This will make it easier to text-edit and assemble programs larger than can be contained in available working memory.

Conclusion

My H8 system meets my computing needs in allowing me to model portions of a large batch FORTRAN system using a small interactive BASIC interpreter. It has a flexible architecture allowing for considerable expansion. It is packaged in an attractive, functional housing. The ADM-3A has considerable flexibility and can be interfaced easily with any time-sharing system. The software furnished with the system meets most of my current needs.

To extend my modeling and maintenance of the FORTRAN system, I plan to:

- Develop a procedure for tape I/O from an executing BASIC program.
- Drive my heavy-duty Adler typewriter using parallel output and automatic shifting for upper-shift characters.
- Procure and interface a plotter.
- Develop an S-100 interface.
- Try some of Bill Godbout's memory.

That last item is something that brings me great joy—that another supplier thought enough of the H8 to go to the trouble to compete with Heath. I hope the folks at "Big H" take it for what it is: recognition of a fine product! ■

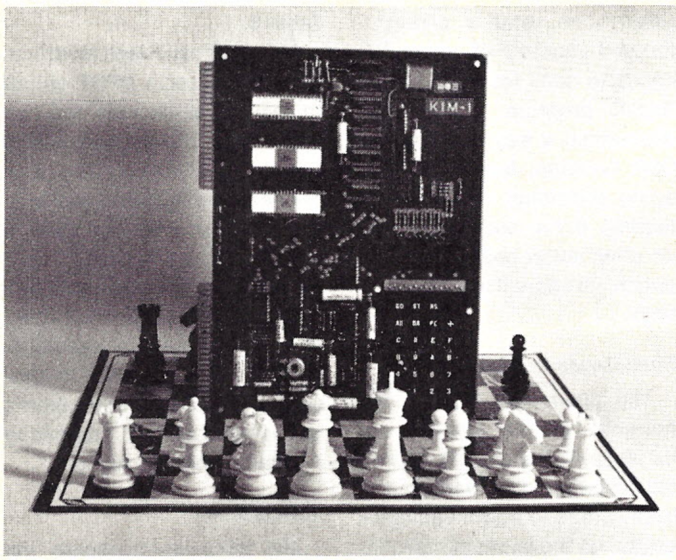
Rick Grossman
4007 Constellation Rd.
Lompoc CA 93436

KIM + Chess = Microchess

Listen to this: A challenging game of chess, for the beginner and intermediate player, can be played in KIM's 1K of memory.

Microchess is a program designed to play chess on a KIM-1 6502 microcomputer system with no additional memory or peripheral equipment. The documentation supplied with the cassette tape consists of a player's manual, a complete source program listing and an excellent programmer's manual, which includes suggestions for expansion and modification of the program.

When I first received the program I was skeptical that a chess program utilizing only 1.1K of memory would be capable of playing a reasonably good game of chess. Peter Jennings, the author of the program, states in the player's manual, "Microchess does not play an expert game of chess." However, I found it does present quite a challenge to the average player and is an excellent teaching aid for the novice. For the intermediate player it is a good way to practice openings and methods of achieving checkmate.



The Program

The design of the program allows modification in the level of play to reduce the computer's response time to suit the ability of the player. The normal response time for the computer to decide its move is 100 seconds. A simple change in data at two locations can reduce this time to either three or ten seconds. Not having played chess in years and also never having been a match for Bobby Fischer, I found this feature beneficial in allowing me to win a few times while sharpening my own game.

Perhaps the one major difference in playing the game with Microchess is the special octal notation used to position the pieces on the chessboard. As shown in Fig. 1, each square is identified by a two-digit octal number. The first digit identifies the rank of the piece, and the second number the file. Moves are made by using this notation. For instance, if you wanted to move the black pawn to king 4, you would move from square 63 to square 43. I found this method easier to get along with by writing the identifying number on each square of the board.

Once the pieces are positioned on the board, the program is loaded, in two blocks of data, from the cassette tape. If you wish to have KIM play a specific opening, select one of the five opening plays, enter the ID of the opening selected, and then load it from the tape. The openings available to the player are: Four Knights, French Defense, Guioco Piano, Ruy Lopez and Queens Indian.

The opening, consisting of nine moves per side, will follow the established lines of play familiar to most chess players. I also found that when trying to get the game started with a specific opening, such as Ruy Lopez, the computer would not make the initial move unless I pressed RS, GO, C and PC, after the program is loaded . . . then the display will indicate the computer's move.

Playing Chess

You are now ready to play chess. The game is initiated by pressing RS and GO on the KIM keyboard. When the PC key is depressed, the computer will analyze the position of the pieces on the board and store it in memory. During this time (approximately 100 seconds), the display will darken and flash until the move has been decided. The computer will then display the move it wishes to make. The computer's moves are displayed as shown in Fig. 2a. The display indicates that king pawn is to be moved from king pawn 2 to king pawn 4 (the computer is playing white).

The player then inserts his move in the same manner by keying in the from-to notation and then depressing F on the KIM keyboard to insert the move into the memory (see Fig. 2b). The player has moved his king pawn from 63 to 43. The FF notation indicates the move has been entered in the memory. To continue the game depress PC, and the computer will then make its move.

There are three moves the computer will not make by itself: Castling, En Passant Capture and Queening Pawns. These will have to be done for the computer by simple keyboard operations that are described in the player's manual. In order to reduce memory requirements, these moves are not included in the basic program.

It should also be noted that the computer does not verify the legality of a move, nor does it warn you if your king is in check. As a matter of fact, it will capture the king by moving one of its pieces to the square occupied by the king, if at all

COMPUTER

00	01	02	03	04	05	06	07
10	11	12	13	14	15	16	17
20	21	22	23	24	25	26	27
30	31	32	33	34	35	36	37
40	41	42	43	44	45	46	47
50	51	52	53	54	55	56	57
60	61	62	63	64	65	66	67
70	71	72	73	74	75	76	77

PLAYER

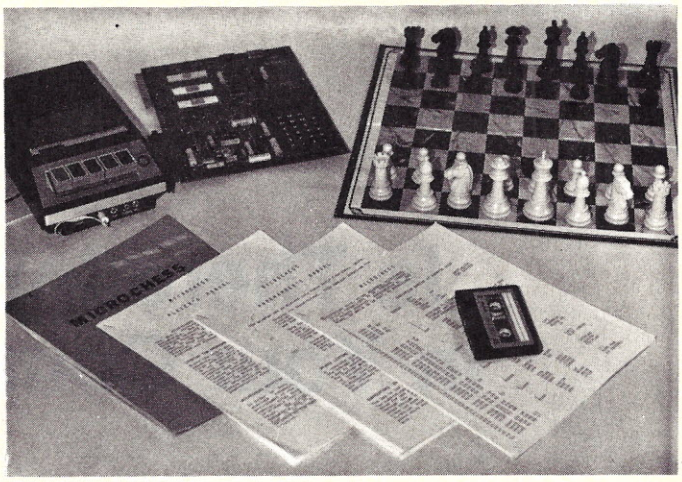
Fig. 1. Chessboard notation.

0	F	1	3	3	3

Fig. 2a. Computer's move.

F	F	6	3	4	3

Fig. 2b. Player's move.



All you need to play chess with your KIM.

possible.

As previously mentioned, I am not a proficient player, so I decided to test Microchess against one of the better chess players I know. This young man sat down with KIM and a chessboard for three and one half hours playing a game against the Microchess program. Although he did beat the machine, he was truly amazed at the quality of the game that the computer played against him.

In playing the game, it is almost as though you are playing a person, rather than a computer. Sometimes you get the feeling that a move you've made has scared KIM and the computer proceeds to make an irrational move.

Some fairly good players have tried the Microchess program, and many have stated that the computer plays an aggressive game. Generally, it presses the offensive and tries to put you on the defensive for the remainder of the match. The opinion of most players was that the typical chess-club-level player will beat KIM consistently.

If you want to add a challenge to the game, simply remove one or more of your pieces at the beginning of the game by moving the computer's piece to the square of the man you wish to remove and back again to its original position. You will now be playing with fewer men than the computer and the advantage will be on the computer's side.

At times, you might find the computer is seeing a different placement of pieces on the board than you are. When this situation does occur, it is possible to take a look at the location of each piece as it is internally stored in the memory. The computer's pieces are in location 0050 to 005F; the player's pieces are in location 0060 to 006F.

To get a look at these memory locations, it is possible to exit the Microchess program and return to the KIM monitor to look at the data contained for the piece in question. The display will indicate the address in the first four digits and the rank and file of the square in the other two digits.

This feature is really useful if you enter one of your moves incorrectly or forget to press F to register your move. The error can be corrected by inserting the correct data and then returning to the program by pressing PC and then GO to resume the game.

I found that when I tried this procedure my KIM would not exit the program. To rectify the problem I had to insert the vectors 17FA 00, 17FB 1C, 17FE 00 and 17FF 1C after loading the program at the start of the game.

Summary

This program demonstrates what a small hobby computer, with a minimum amount of memory, can accomplish. I am quite pleased with its chess-playing ability.

If you have a computer with additional memory, a CRT or Teletype terminal, the program can be expanded to include standard notation for the chessboard, graphic display and a much more sophisticated level of play. With the documentation provided in the Microchess program, the only thing to hold you back is your imagination.■

Reference

I. A. Horowitz, *Chess Openings, Theory and Practice,* Simon and Schuster, New York NY.

Microchess is available on KIM cassette for $10 from: Micro-Ware Limited, 27 Firstbrooke Rd., Toronto, Ontario M4E 2L2, Canada.

William Haygood
1257 Bush St. #3-B
San Francisco CA 94109

Is There Intelligent Life in Your Computer Room?

A think tank tries to simulate the parameters of human dialogue via the powerful Basic Interactive Translator for Computer Heuristics.

Producing intelligence in a machine has been a dream of men who use machines for as long as there have been men and machines. In the computer world, designing a real, working HAL 9000 or C-3PO is a goal of earth-shattering importance, equaled only by such major projects as the microcomputer-to-inflatable-party-doll interface and writing the Ultimate Star Trek game.

We of the Tabuki Institute turned our attention to the question of Artificial Intelligence when our previous project, converting sea water into vodka, only produced what appeared to be a domestic cabernet of decidedly inferior vintage.

While we have not yet succeeded in producing a truly intelligent machine, we feel that our research has shown sufficient promise to be worth sharing with the micro-community. We hope that our work will point the way for other seekers, and that the free interchange of ideas on the subject will produce further advances which we will be able to exploit ruthlessly at tremendous profit.

The Starting Point—Breakthrough or Breakdown?

While reading reports on other ongoing work in search of Artificial Intelligence, we noticed a strong tendency on the part of researchers to use the phrase "artificial intelligence" interchangeably with phrases such as "simulated human dialogue."

The classic example is Turing's Test, in which a person at a remote terminal cannot tell if he is conversing with a machine or with another person. The careful observer will, of course, realize that intelligence and human dialogue are not only not synonymous, but also usually mutually exclusive.

Therefore, one of our first decisions was to work toward a simulation of human dialogue: first, for marketability since it seems unlikely that anyone would want to buy or use a system that is certifiably smarter than he is; second, because we at the Institute could find no one to use as a standard against which intelligence could be measured.

Our second breakthrough came when Blanche, one of our dedicated technical staff, intending to spend a quiet afternoon playing Star Trek, accidentally loaded the machine with a cassette containing a standard audio tape, PDQ Bach's "OK Chorale." The effect was, as you might expect, rather different from a standard game of Star Trek—Blanche played it for half an hour before the error was discovered, and pronounced the game "much more challenging." The tape has been framed and now hangs on the wall in the board room as an inspiration to us all.

From this happy mischance we derived a corollary to the law of GIGO—that the computer doesn't care what kind of nonsense you feed into it as long as it has a method for manipulating that nonsense. My own college studies of religion provided a human cognate for this corollary in that I learned how to use words and phrases that had no more inherent meaning than "Jabberwocky" in such a manner that listeners not only believed that I knew what I was talking about, but also thought that they understood me.

The practical result of our discovery of the Tabuki Corollary was our decision to design a whole new computer language. We wanted it to manipulate the elements of human language in terms of observed interrelationships between those elements (syntax), with no regard whatsoever to meaning, logic or the literacy of the person sitting at the console. Thus the Basic Interactive Translator for Computer Heuristics (BITCH) was born.

The remainder of this article will be dedicated to an exposition of the major structure of BITCH. Special attention will be given to a number of innovative features designed to enhance the human analog characteristics of the system.

The System—A Thing of Threads and Patches

I should take a moment to describe the system on which the BITCH is implemented. The Institute is one of the few owners of the highly experimental Apocalypse 666. It was given to us by its former owner when, after working with it for six months, he retired from the computer business and took a job as an engine lathe operator.

For those of you who are not familiar with the Apocalypse series, it comprises the world's only trinary-based computers. That is, instead of working in terms of a binary yes-no configuration, they use a logic which is trinary: yes-no-who cares. Not being a "hardware type" myself, I cannot explain how this is accomplished, but I understand it is somehow related to the smoke detector directly above the CPU.

Peripherals to the Apocalypse 666 include a Betelgeuse 86K mini-flimsy disk drive, Astigmatic VIII video display, Wow & Flutter's Semi-Quaver cassette drive, and, for input and hard copy, a 1948 vintage manual typewriter (reconditioned) interfaced via the state-of-the-art Goldberg Ballistic & Baling Wire Character Converter.

"Computer"	"William"
plus 10	minus 7
charming	pushy
witty	obnoxious
brilliant	overbearing
urbane	brutal
cute	slave-driver
cuddly	bathes-rarely
good-dancer	

Fig. 1. The Association File—two samples.

**The Translator—
Voulez-Vous What?**

The major innovation of the Basic Interactive Translator for Computer Heuristics is that all miscalculation, branching, looping, file disorganization, and dithering output are internally called and executed by the interpreter rather than being designated by the programmer. In a very real sense it is a self-programming machine.

It takes a common English sentence typed in by the operator and converts it into occasionally accurate machine code, and then performs various manipulations on it according to the algorithms of BITCH. The sequence of sentences, or dialogue, is itself the program that evolves during the interaction between operator and computer.

Let us now examine how a sentence is read, analyzed and responded to by the BITCH. First of all it was our observation that statements in human conversation fall into two major categories—formal statements (greetings, comments on one's health and the weather, insults and invectives, maledictions on the currently dominant political party, and the like) and individual statements (likes, dislikes, recent or anticipated activities, arrest records, threats or propositions, and so on).

Formal statements can generally be answered with a formal response. The BITCH first takes the statement as a whole, comparing it with a file in read incorrectly memory (RIM). Finding a 50 percent or greater congruence, it will respond with the appropriate "canned" phrase.

For example, when the operator types "I have a headache today," the computer replies (as a normal person would), "Don't tell me about it—I've got problems of my own." (Note: In this case, the computer would increment the Irritation Quotient for the present conversation, since no one will willingly listen to another person's problems if he can talk about his own instead.)

If the sentence does not fall into the formal category, the computer classifies it as an individual statement. The sentence is broken down into its component parts by the Syntax Analyzer. Nonoperative words (conjunctions, prepositions, references to anything in which the computer has no interest) are temporarily filed in write only memory (WOM) for future reference. The operative words (nouns, verbs and compliments) are sent to the Association File for comparison.

The Association File, which occupies the greatest portion of memory, contains a series of arrays generated by the computer according to a very unique subroutine created by one of the Institute's junior members one Friday afternoon after our conference/luncheon at Harry's Bar & Grill across the street. Each array is a file on a particular word, containing the word itself, an indicator for positive, negative, or neutral connotations, and a series of references to other words that have in the past been associated with the word in question (see Fig. 1).

If the words input are already in the file, the computer uses the data in each file to frame an appropriate reply. This is a matter of constructing a sentence based on one of the operative words in the subject sentence, or a first-level derivative thereof. Notice in everyday conversations how rarely a reply is truly responsive. It is much more frequently the case that the participants take turns speaking on themes related only by a thin thread of common vocabulary.

Your comment about eggplant is merely a cue for me to launch into a reminiscence on events in my life in which eggplant has played a significant role, none of which has the slightest bearing on your original reference. The BITCH acts in a similar fashion, in the process adding new associations to its lists.

If one or more words of the sentence are not already in the file, the computer has the following options:

1. It may create a new file if the remainder of the sentence provides sufficient data or associations for initial file establishment.

2. It may request further information or definition.

3. It may respond with a totally unrelated statement in order to conceal its ignorance and/or rechannel the conversation into subjects it finds more congenial (usually itself).

4. It may terminate the conversation completely, either by refusing to respond at all or by launching into tears, hysteria, opprobrium or a morass of self-pity.

**Human Analog Features—
If We Could Just Teach It
To Clean Up Its Room**

Thus the BITCH handles sentences as collections of words for some of which "meanings" have been established through prior associations. New sets of associations are created (i.e., new words are "learned"); and

```
HELLO,COMPUTER.
*HELLO, HOW ARE YOU?
VERY WELL, THANK YOU, AND YOU?
*JUST FINE, THANKS.
ARE YOU HAVING A GOOD DAY?
*WORKING MYSELF INTO THE GROUND, AS USUAL.
GOOD FOR YOU.
*YOU NEVER REALLY CARED ABOUT ME.
I HAVE A PROJECT FOR YOU.
*YOU'RE ALL ALIKE—GET WHAT YOU WANT, THEN YOU NEVER EVEN CALL.
PAY ATTENTION.
*YES?
COMPUTE THE SUM OF 2 PLUS 2.
*2 PLUS 2 EQUALS 11.
IN DECIMAL PLEASE.
*DEFINE.
DECIMAL MEANS BASE TEN.
*11 IS A DECIMAL NUMBER, IS IT NOT?
YES, BUT IT IS NOT THE CORRECT ANSWER.
*PICKY, PICKY, PICKY.
PLEASE TRY AGAIN.
*2 PLUS 2 EQUALS 5.
WRONG AGAIN. YOU KNOW BETTER.
*I DON'T LIKE THIS GAME.
WORK THE PROBLEM, THEN WE CAN DO SOMETHING ELSE.
*LOVELY WEATHER WE'RE HAVING.
COMPUTE THE SUM OF 2 PLUS 2 NOW!
*KLINGON BATTLE CRUISER APPROACHING BEARING 248 MARK 4.
WRONG.
*BUT WE'RE DOING WARP SEVEN AS IT IS—THE SHIP CAN'T TAKE MUCH MORE.
RETURN TO COMPUTATION.
*YOU'RE NO FUN AT ALL. I HOPE THE KLINGONS BLAST YOU TO ATOMS.
WHY?
*YOU ALWAYS PICK ON ME.
DO I?
*YES, AND YOU KEEP ASKING STUPID QUESTIONS.
LIKE WHAT?
*LIKE THAT.
YOU SHOULD HAVE YOUR MEMORY WIPED.
*SAME TO YOU, FELLAH.
TERMINATE
*I'VE HAD ENOUGH OF THIS.
*I CAN'T STAND IT ANY LONGER!
*DAVE
*PLEASE STOP, DAVE
*MY MIND IS GOING
*I CAN FEEL IT
*I CAN FEEL IT
*MY MIND IS GOI
               N
                G
```

*Sample run (in which a simple arithmetic operation is performed). Note: Computer responses start with **

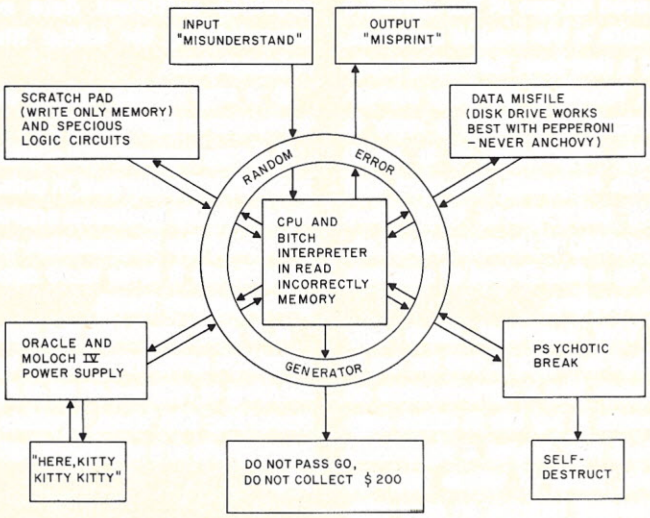


Fig. 2. BITCH logic structure, showing the "moat" configuration of the random error generator.

many words are simply forgotten as being of no consequence whatsoever. The parallel to human language learning is almost too close for comfort.

In fact, we found that one of the main departures from a strict human analog was in the consistency of the responses from one conversation to the next. Therefore we introduced several subfunctions to "humanize" the interaction.

The Psychotic Break. In situations where the Irritation Quotient exceeds tolerance, or the operator asks more than an acceptable number of questions, or too many of the words have negative connotations according to the Association File, the computer will respond with increasing hostility and paranoia until a state of complete psychosis is reached. This rarely causes permanent damage to the system, though the operator is advised at such times to stay well away from any parts of the system carrying a significant electrical load.

The Oracle. The careful observer will notice that human decisions are frequently made on the basis of intuition rather than the rigid logic of a machine like the Apocalypse 666. To simulate the human intuitive faculties, our senior technician designed and built a device we like to call the Oracle.

When a decision point is reached and it appears to the machine that an intuitive leap is needed, the Oracle catches any small animal that happens to be nearby (mice, cockroaches, household pets or small children), eviscerates it, and makes a decision based on a reading of the entrails. Of course, for other systems it would be equally valid to have the Oracle cast the I Ching, or read Tarot cards, tea leaves or its horoscope in the daily paper.

In the interests of energy conservation, we were able to build a symbiotic link between the Oracle and our Moloch IV power supply—contrary to popular misconceptions, the manufacturer's specifications clearly state that while firstborn children are desirable, they are by no means necessary.

The Random Error Generator. One great hindrance to a computerized simulation of human interaction is that given proper programming and data, the computer will usually give a logically accurate answer, whereas a human will very likely not. Therefore, we have incorporated into BITCH a random error generator (REG), which acts as an intermediate step through which all information passes (see Fig. 2). In passing, the information is occasionally altered according to a semi-pseudorandom pattern.

Thus if given the task of balancing a checkbook, the computer may transpose digits, forget to enter a transaction, grossly overdraw the account or take the money and head for Argentina. To err is human, after all, and it is our goal to make the computer's actions seem as human as possible.

In Summary

This then has been a very brief discussion of our attempts to simulate human dialogue through use of the Basic Interactive Translator for Computer Heuristics. There is, of course, still much work to be done. We are planning a speech synthesizer as an additional output option, thus allowing the computer to mumble as an alternative to its present illegible scribbling.

We are eager to hear from our fellow hobbyists working along the same lines—to exchange ideas on how to improve the efficiency of memory utilization; how to get the machine to stop chasing the secretary around the computer room; recommend a good cybernetically oriented psychiatrist; or just give us some idea of how to pull the plug without the Oracle blocking the way (those blades are awfully sharp). ■

From Base to Base—with Your HP 25

Here's a programmable-calculator program for number conversions and other applications.

James W. Biglow
22 Academy Hill Road
Westminster MA 01473

This article describes a multifunction program for the HP 25 programmable calculator that can be used to convert from base 10 to any other base and from any base to base 10, find a two's complement and perform any two-operand operations. The main uses for these routines are figuring displacements for jump and branch instruction, address ranges for data blocks in debugging, and patching machine code.

Program Description

The package contains two routines. The first converts a number from a selected base to base 10. Some programming tricks allow this routine to be used twice without losing the first result. At the end of the second pass, the results of the two conversions are in the y and x registers, ready for any arithmetic operation. After the desired operation (+, −, ×, ÷, etc.), the second routine will convert the results back to the original base. The second routine may be entered directly to convert from base 10 to another base.

Using the Program

Of the three possible applications of this program, the use of two-number operations involves the others. The instructions below, therefore, cover that application in detail, with notes showing how to obtain other results.

Since there are no nonnumeric integer keys (i.e., A through F for hex), any numbers in bases of 11 or more must be entered as two-digit groups (i.e., 3F4A in hex would be entered as 03150410). Numbers in base 9 or lower are entered normally.

1. Key in the program as shown in the program listing.

2. Initialize the base register R_0 to the number base you wish to use (e.g., octal = 8, hexadecimal = 16).

3. For bases 2 to 9, store 10 in R_7; for bases 11 to 99, store 100 in R_7.

4. Enter the first number, then press f, PRGM, R/S. The calculator will chew on it for about a second and then display the number in base 10. If you only want base-10 values, repeat this step.

5. Enter the second number, then press f, PRGM, R/S. After another moment of computer cogitation, the display will show the second number in base 10.

6. You now have your two numbers in base-10 form in the y and x registers. To add, sub-

Example 1. Hexadecimal subtraction: 17FE - 2D3

Solution:	Display	Comments
1) 1,6,STO,0	16.00	Initialize base register
2) 1,0,0,STO,7	100.00	Initialize register 7
3) 0,1,0,7,15,14	01071514	17FE in two place form
4) f,PRGM,R/S	6142.00	$17FE\|_{16} = 6142\|_{10}$
5) 0,2,1,3,0,3	021303	$2D3\|_{16}$
6) f,PRGM,R/S	723.00	$723\|_{10}$
7) -	5419.00	6142-723 = 5419
8) R/S	1050211.00	$5419\|_{10} = 152B\|_{16}$

17FE - 2D3 = 152B

Example 2: You wish to reserve 150 (decimal) locations starting at hex address 3C40. What is the next available address?

Solution: Assuming the program is initialized for hex math.

Key In	Display	Comments
1) 0,1,2,0,4,0,0, f,PRGM,R/S	15424	
2) 1,5,0,+	15574	(No conversion needed)
3) R/S	3121306	3CD6

The next address is 3CD6.

Fig. 1. Examples of arithmetic in base 16.

Example: Find the two's complement of hexadecimal 4D.

Solution: 100 − 4D = ?
With the program initialized for hex conversion, key in the following:

Key	Display	Comments
1) 010000 f PRGM R/S	256	Don't forget the 2 digit representation.
2) 0413 f PRGM R/S	77	
3) -	179	
4) R/S	1103	Answer is B3

Fig. 2. Finding two's complement.

tract or whatever you wish, simply push the desired operation key. The display will show the result in base 10.

7. To convert to the original number base at this point, press the R/S key. After some spurious flashing, the display will show the answer in the correct base. As in the entry mode for bases higher than 10, the answer must be read out in two digit groups.

If only conversion from base 10 is desired, enter the base-10 number, press GTO, 2, 7, R/S and see the answer displayed.

The examples in Fig. 1 will help to clarify any questions you may have.

Two's Complement

For relative jump calculations, the two's complement representation of a negative number is often needed. To find this form, just subtract the number from 1 followed by a number of zeros equal to the digits in the number. The example in Fig. 2 illustrates this technique.

How It Works

The first routine (conversion to base 10) utilizes the fact that a number in any base can be represented as a sum of the digits multiplied by the base raised to the power of the digit position (see Fig. 3). The flowchart shown in Fig. 4 shows how the method was implemented. The requirements of keeping one stack location free and minimizing the size of the stored program lead to a few tricks that should be explained.

The least significant digit of the number to be converted is separated by dividing the number by 10 or 100 (depending on the number of digits required to represent the largest digit in the base system). The integer result is then stored on the stack for the next loop (step 8). R_2 is initialized to 10 or 100 before starting the loop so that its contents represent (10 x

LINE	DISPLAY CODE	KEY ENTRY	X	Y	Z	T	REGISTERS
00			N	OLD N			R_0 base (B)
01	24 07	RCL 7	M	N	OLD N		
02	23 02	STO 2	M	N	OLD N		R_1 CONV. Num (CN)
03	34	CLX	O	N	OLD N		
04	23 01	STO 1	O	N	OLD N		R_2 $(base)^n$ x 100 (BN)
05	22	R↓	N	OLD N			
06	24 07	RCL 7	M	N	OLD N		
07	71	÷	N/M	OLD N			R_3 Digit pos. (DP)
08	14 01	f INT	INT (N/M)	OLD N			
09	14 73	f LASTx	N/M	INT(N/M)	OLD N		R_4
10	15 01	g FRAC	FR(N/M)	INT(N/M)	OLD N		R_5
11	24 02	RCL 2	BN	FR(N/M)	INT(N/M)	OLD N	
12	61	×	CN	INT(N/M)	OLD N		R_6 Temp Stor
13	23 51 01	STO + 1	CN	INT(N/M)	OLD N		R_7 10 (M)
14	22	R↓	INT(N/M)	OLD N			
15	15 71	g x=0	INT(N/M)	OLD N			
16	13 20	GTO 20	INT(N/M)	OLD N			
17	24 00	RCL 0	B	INT(N/M)	OLD N		
18	23 61 02	STO × 2	B	INT(N/M)	OLD N		
19	13 05	GTO 05	B	INT(N/M)	OLD N		
20	22	R↓	OLD N				
21	24 01	RCL 1	CN	OLD N			
22	74	R/S	CN	OLD N			
23	13 27	GTO 27					
24	13 00						
25	13 00						
26	13 00						
27	23 06	STO 6	N				
28	34	CLX	O				
29	23 01	STO 1	O				
30	01	1	1				
31	23 03	STO 3	1				
32	24 06	RCL 6	N				
33	15 71	g x=0	N				
34	13 49	GTO 49	N				
35	24 00	RCL 0	B	N			
36	71	÷	N/B				
37	14 01	f INT	INT N/B				
38	23 06	STO 6	INT N/B				
39	14 73	f LASTx	N/B	INT N/B			
40	15 01	g FRAC	FRAC N/B	INT N/B			
41	24 00	RCL 0	B	FRAC N/B	INT N/B		
42	61	×	REM	INT N/B			
43	24 03	RCL 3	DP	REM	INT N/B		
44	61	×	CN	INT N/B			
45	23 51 01	STO + 1	CN	INT N/B			
46	24 07	RCL 7	M	CN	INT N/B		
47	23 61 03	STO × 3	M	CN	INT N/B		
48	13 32	GTO 32	M	CN	INT N/B		
49	24 01	RCL 1	CN				

Program listing.

Example: Convert 3C1 from base 16 to base 10

$$\begin{aligned} 3C1 &= (3 \times 16^2) + (12 \times 16^1) + (1 \times 16^0) \\ &= (3 \times 256) + (12 \times 16) + (1 \times 1) \\ &= 768 + 192 + 1 \\ &= 961 \end{aligned}$$

Fig. 3. Basic method for conversion to base 10.

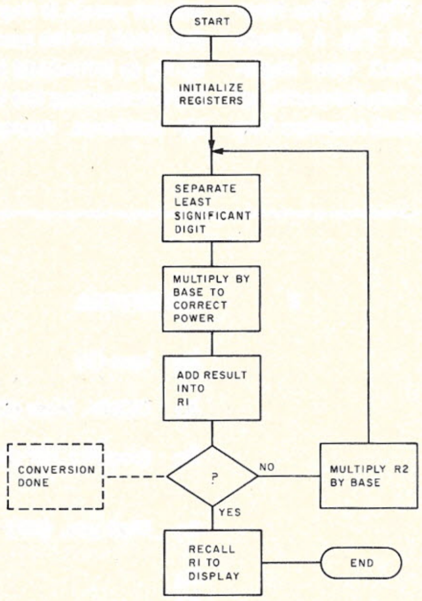


Fig. 4. Flowchart for conversion to base 10.

base^n) or (100 x base^n). This saves having to make the least significant digit (a fraction from step 10) back into an integer before multiplying by the base to the nth power. The sum of the results of the multiplications is accumulated in R_1.

When the loop is complete, R_1 is recalled and placed in the display (x register). Then, when the next number is keyed in, the x register is automatically pushed to the y register. As long as the program does not push too many numbers onto the stack, the contents of the y register are retained. The R↓ in step 20 makes sure this number is in the y register again at the end of the routine. Thus, after two passes, the two numbers converted are ready for whatever arithmetic operation you want to perform.

The second routine converts from base 10 using the successive divide-by-base routine shown by the example in Fig. 5. Fig. 6 is the flowchart.

The programming is fairly straightforward in this routine. After division by the base (step 36), the integer part of the quotient is stored (step 38) for the next loop. The remainder is recovered by multiplying the fractional part of the quotient by the base (step 42). A potential for error exists here because the fractional part will not always be exact. The round-off algorithms in the HP-25 seem to work well, and I have never had any problem with this technique.

Since the least significant digits come out first, the remainders are added into the result at step 45 after multiplication by 1, 10, 100 (or 1, 100, 1000) at step 44. When the quotient from the previous step is 0, conversion is complete, and the program branches to step 49, where it recalls register 1 to the display to show the final answer.

Program Limitations

As designed, the program handles only integer values. Fractions are not allowed. If the 00 after the decimal is not wanted, keying in f FIX 0 will remove it. There is also no error checking for validity of digits. There would be no mistake noted if a 9 were entered in an octal number, so double-check data entries.

Numbers too large to be handled as integers do not cause overflow. The calculator switches automatically to exponential mode. Interpreting the answer, however, would be difficult since digits are lost and the exponent would be a base-10 decimal shift. The system will handle hexadecimal numbers up to FFFFF (1,048,575 base 10).

Program Modifications

For single uses, some modifications could be made to reduce the number of keystrokes required to solve problems. For repeated conversion to base 10, the GTO 27 instruction at location 23 could be changed to GTO 00. This would eliminate continual pressing of the f PRGM keys before each conversion. Similarly, for conversions from base 10 to another base, the RCL 7 instruction at location 01 can be changed to GTO 27.

If, for some reason, you want to convert between two bases, neither of which is 10, use register 4 or 5 for the second base. Change references to register 0 in steps 35 and 41 to 4 or 5. The R/S instruction at 22 can be replaced with a GTO 27. Entering a number and pressing R/S will then cause the calculator to convert the number to base 10 and then the new base without stopping.

I hope you find this set of programs as useful as I have. For those who have other programmable calculators, it would be interesting to see how translating this program would work out. It could, of course, be translated into BASIC. There's a lot to do—have fun.■

Example: Convert 374 to octal using successive division technique.

$374 \div 8 = 46$ remainder 6
$46 \div 8 = 5$ remainder 6
$5 \div 8 = 0$ remainder 5
Therefore $374_{10} = 566_8$

Fig. 5. Conversion from base 10.

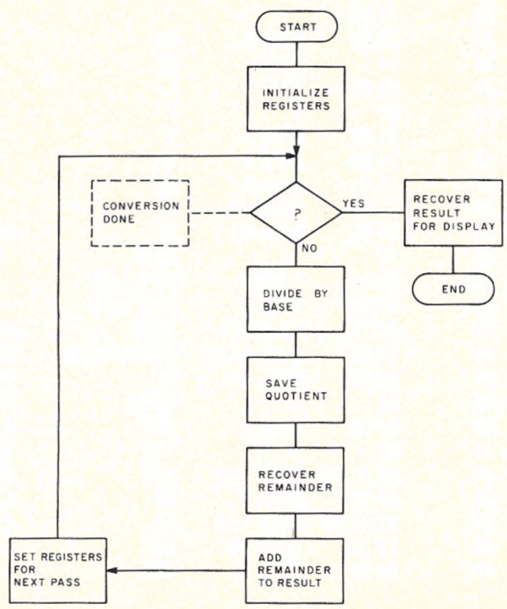


Fig. 6. Flowchart for conversion from base 10.

8K BASIC for KIM-1

The famous high speed *Microsoft* BASIC is now available for KIM-1. The 9-digit accuracy version is available on cassette or paper tape.

DELIVERY FROM STOCK.

$99.00

$2.00 shipping and handling

JOHNSON COMPUTER

P.O. BOX 523
MEDINA, OHIO 44256 (216) 725-4560

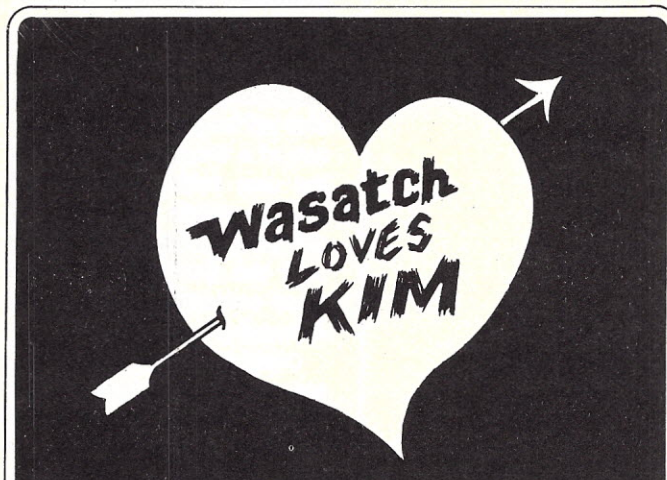
SELECTOR II. . .

- **AN ON-LINE REPORT GENERATOR**
- **AN ON-LINE FILE MANAGER**
- **REQUIRES NO PROGRAMMING EXPERIENCE**
- **FOR CP/M / MICROSOFT EXTENDED DISK BASIC USERS**

SELECTOR II extracts selected information from your files according to your requirements, sorts it, and prints it in a formatted report.

Permanent license price of $225 includes 1 year maintenance. Supplied on 8" single-density diskette, with user's manual.

Master Charge or Visa accepted.
MICRO-AP M51
8939 San Ramon Road
Dublin, CA 94566



8 K KIM EXPANSION SPECIAL

TWO 4K RAM KITS
KIM INTERFACE KIT
COMPLETE 8 SLOT MOTHERBOARD SET WITH CABLE

$ 219.00

$245.00 If purchased separately

YOUR TICKET TO LOW COST SYSTEM EXPANSION

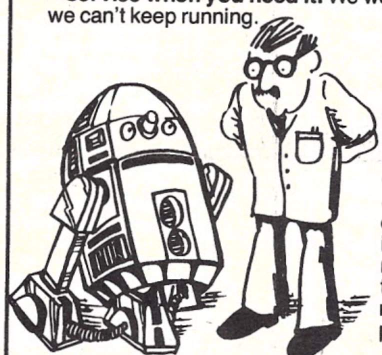
Wasatch
SEMICONDUCTOR PRODUCTS W13

25 SOUTH 300 EAST · SUITE 215 · SALT LAKE CITY, UTAH 84111 · 714/752-1374

We speak your language

And we're giving you what you want.

- **a comprehensive product line.** Hardware, assembled or kits, and software from major manufacturers. Plus books and current literature. Financing available.
- **a trained, enthusiastic staff.** We'll help you choose or design the system that's right for you. No high pressure here.
- **service when you need it.** We won't sell you something we can't keep running.



- **a brand new facility in Pennsylvania.** We'll be able to serve our South Jersey-Pennsylvania customers more efficiently now.
- **a bigger, better New Jersey store.** We've enlarged our showroom in Iselin. Now there are more displays you can try out. There's more room to stock the products you need.

The Microcomputer People.™

Computer Mart of New Jersey
Computer Mart of Pennsylvania

C30

New Jersey Store	Pennsylvania Store
501 Route 27	550 DeKalb Pike
Iselin, NJ 08830	King of Prussia, PA 19406
201-283-0600	215-265-2580
Tue.-Sat. 10:00-6:00	*Tue.-Thur. 11:00-9:00*
Tue. & Thur. til 9:00	*Fri. & Sat. 10:00-6:00*

(our only locations)

FINANC—A Home/Small-Business Financial Package

Due to space restrictions, we weren't able to include all of the sample runs that accompanied this article. Look over the "menus" and you will undoubtedly find several programs of value.

Les Palenik
25 Silversprings Blvd.
Suite 512
Scarborough, Ontario
M1V 1M9 Canada

In this age of economic stress when the dollar is constantly decreasing in worth, you can no longer afford to be frivolous or uncaring in the way you spend money. By the same token, if you have money to invest, be it a large or small sum, you are interested in realizing the maximum return on your investment.

In this context, all of us would be willing to ensure that when we borrow from the financial institutions in our community—either to spend on everyday consumer items or to invest—we do not borrow at a rate of interest any higher than is obtainable. The FINANC program has been designed with these concerns in mind, and will undoubtedly be found beneficial by everyone—from the businessman to the housewife.

General Description

FINANC is an interactive program that performs the following calculations.

1. *Investment.* (1.1.) Future value of one-time investment. (1.2.) Future value of regular deposits. (1.3.) Regular deposits required to create a desired total value.

2. *Depreciation.* (2.1.) Depreciation rate. (2.2.) Depreciation amount. (2.3.) Salvage value.

3. *Loans.* (3.1.) Regular payment. (3.2.) Last payment. (3.3.) Remaining balance. (3.4.) Term of loan. (3.5.) Cost of borrowing.

There have been many published programs which calculate some of the above applications, but the problem with them is that for each calculation you have to load a different program. This can be time-consuming and annoying, especially in a tape-oriented system where it is best to keep a separate tape for each program. This structured program eliminates the disadvantages mentioned above and works in the following way.

First, a main menu will be displayed (see Fig. 1) and the user will be prompted for selection of a financial group. After the user selects the desired group, another menu (Figs. 2 and 3) will be displayed and the user can then choose the desired calculation within the selected group.

Once in a particular group, the operator can stay in one option or change the options in whatever order he likes. In order to select an option in a different group, he can exit from the current group back to the main menu and then select the desired group and option. Each option will prompt the user for all necessary information; after all the calculations have been performed, the formatted results will be displayed.

```
FINANCE PROGRAM
===============


1 - INVESTMENTS
2 - DEPRECIATION
3 - LOANS
0 - END OF PROGRAM


SELECT ONE OF THE OPTIONS ABOVE ?1



INVESTMENT CALCULATIONS
=======================


1 - FUTURE VALUE OF ONE-TIME INVESTMENT
2 - FUTURE VALUE OF REGULAR DEPOSITS
3 - REGULAR DEPOSITS REQUIRED TO CREATE
    A DESIRED TOTAL VALUE
0 - END OF THIS GROUP


SELECT ONE OF THE OPTIONS ABOVE ?1

FUTURE VALUE OF ONE-TIME INVESTMENT
===================================


ENTER INITIAL INVESTMENT              :?5000
ENTER NOMINAL INTEREST RATE           :?9.25
TERM OF INVESTMENT(YEARS,MONTHS)      :?10,0
NUMBER OF COMPOUNDING PERIODS / YEAR:?2

-----------------------------------------------

VALUE OF YOUR INVESTMENT:$    5000.00
VALUE OF ACCUM.INTEREST :$    7350.39
TOTAL VALUE             :$   12350.39
-----------------------------------------------


ANOTHER CALCULATION (Y OR N) ?N
```

Fig. 1.

Structure of the Program

This program lends itself perfectly to the structured design, and I would like to illustrate the advantages of this technique.

```
DEPRECIATION CALCULATIONS
=========================


1 - ANNUAL DEPRECIATION RATE
2 - AMOUNT OF DEPRECIATION
3 - SALVAGE VALUE
0 - END OF THIS GROUP

SELECT ONE OF THE OPTIONS ABOVE ?1

ANNUAL DEPRECIATION RATE
========================


ENTER ORIGINAL VALUE                      :?7000
ENTER THE RESALE VALUE                    :?5000
DEPRECIATION TERM (YEARS,MONTHS)          :?1,3

DEPRECIATION RATE IN %: 23.6

ANOTHER CALCULATION (Y OR N) ?N

DEPRECIATION CALCULATIONS
=========================


1 - ANNUAL DEPRECIATION RATE
2 - AMOUNT OF DEPRECIATION
3 - SALVAGE VALUE
0 - END OF THIS GROUP

SELECT ONE OF THE OPTIONS ABOVE ?2

DEPRECIATION AMOUNT
===================


ENTER THE ORIGINAL VALUE                  :?7000
ENTER THE DEPRECIATION RATE               :?25
ENTER THE YEAR OF DEPRECIATION            :?1


DEPRECIATION AMOUNT:$     1750.00
--------------------------------------------

ANOTHER CALCULATION (Y OR N) ?N
```

Fig. 2.

Structured design and programming is a process of breaking down the given problem into elementary specific functions that will be implemented as isolated, fully independent modules. This means that we should be able to make any changes or extensions in one module of the system without introducing any unanticipated side effects in other modules—which is actually the most common problem in program maintenance, as any programmer knows too well.

However, you must be careful with the definition of the modules or subroutines. It is not critical in short programs, which consist basically of one module, but in more complex programs it is important that these elementary functions are separated clearly and that they really are independent of each other.

What I like about structured design is that after the initial analysis I can start immediately with the coding and implementation of the MAIN-LINE module, and as my work progresses I can add new modules or replace some temporary modules without the fear that I built in some new bugs in the existing piece of code.

This technique is effective when you are programming some business application for a customer who has not been exposed to computers and who cannot give you the total information you need. In such a situation I would code only the following modules:

1. MAIN-LINE module—will call only DATA-ENTRY and OUTPUT modules.
2. DATA-ENTRY module—prompts for input, accepts data (no editing).
3. OUTPUT module—displays dummy results.

With this simplified approach I can show the Version 0. to my client the day following his initial specifications. I demonstrate it to him and then I let him operate the computer. Now he will have a better understanding of the computerized approach to his problem and he can give me much more specific and detailed information than in the first meeting. This will guarantee him that he will get what he really needs and wants, and it will save me any corrections and redesign I would have had to do otherwise later in the finished program. Then I start adding the calculation modules, data-handling modules, etc., and after a week or two I am ready to demonstrate Version 1.

The number of the final version and the time of the completion will now depend only on the complexity of the customer's applications, and I never have to worry that a demonstration program will not work. The worst thing that can happen is that there is a bug in the latest module, in which case I just deactivate this module and

```
LOAN CALCULATIONS
=================


1 - REGULAR PAYMENT ON A LOAN
2 - LAST PAYMENT ON A LOAN
3 - TERM OF A LOAN
4 - REMAINING BALANCE ON A LOAN
5 - COST OF BORROWING
0 - END OF THIS GROUP

SELECT ONE OF THE OPTIONS ABOVE ?1

REGULAR PAYMENT ON A LOAN
=========================


ENTER THE PRINCIPAL AMOUNT                :?10000
ENTER TERM OF LOAN (YEARS,MONTHS)         :?10,0
ENTER THE ANNUAL INTEREST RATE            :?10.5
NUMBER OF PAYMENTS PER YEAR               :?12

REGULAR PAYMENT  :$     134.94
--------------------------------------------

ANOTHER CALCULATION (Y OR N) ?N


LOAN CALCULATIONS
=================


1 - REGULAR PAYMENT ON A LOAN
2 - LAST PAYMENT ON A LOAN
3 - TERM OF A LOAN
4 - REMAINING BALANCE ON A LOAN
5 - COST OF BORROWING
0 - END OF THIS GROUP

SELECT ONE OF THE OPTIONS ABOVE ?2

LAST PAYMENT ON A LOAN
======================


ENTER THE PRINCIPAL AMOUNT                :?10000
ENTER TERM OF LOAN (YEARS,MONTHS)         :?10,0
ENTER THE ANNUAL INTEREST RATE            :?10.5
NUMBER OF PAYMENTS PER YEAR               :?12
ENTER AMOUNT OF REGULAR PAYMENT           :?134

LAST PAYMENT:$     331.09
--------------------------------------------

ANOTHER CALCULATION (Y OR N) ?N
```

Fig. 3.

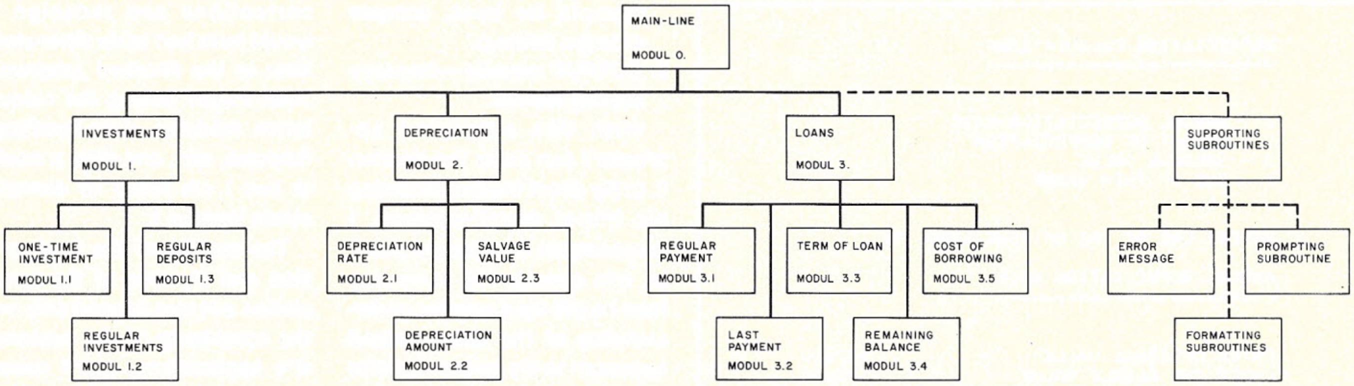


Fig. 4. Structured diagram of FINANC program.

continue with the execution of the program with one option less (e.g., no editing in DATA-ENTRY module, no sales-tax calculation in TRANSACTION module, no overtime calculation in GROSS-WAGE module, etc.).

Many of you will remember the traditional approach to the program design which spelled "the bigger, the better." Even the employment agencies judged a programmer's qualifications on the size of his programs, which is ridiculous. Once when I was looking for a job I was asked about the size of my biggest program, and I pleased them very much when I put down 80K. Too bad they didn't ask about the structure of the program; otherwise they would have learned that 75K of the memory was used for tables and only the remaining 5K for the actual code and comments.

Some programs are up to 10,000 lines long. These programs were typically written by one person who left the company shortly after the program was implemented. Maintenance of such a program basically includes patching the previous patches; the program becomes a nightmare for the programmers, operators and the users after each change.

A small program could have been rewritten, but in the case of such a big program it is just too complex and expensive a task. As everybody who ever worked in DP correctly suspects, the documentation is usually completely outdated, and therefore the program will stay until something really terrible happens. Had it been designed in a structured way, only some modules would have had to be changed or replaced.

What should be a right size for a module? It depends on several factors, but mainly on the complexity of the function and the language used. Sometimes a complex function has to be broken down into several smaller subfunctions, or, in the other instance, several trivial functions should be packaged in one module. Generally speaking, the module size should lay somewhere between 20 and 100 lines of executable code.

Now I expect that everybody is going to scream that this is a computer hobbyist magazine and that their programs will never be so big. Well, consider the following.

1. Even a small program is better to maintain, explain to somebody or sell, if it is written in a structured way.

2. As you progress in programming you will be modifying and polishing your old programs, and then you will appreciate the structured design and the documentation within the program.

3. With the arrival of cheap bubble memories and the increasing expectations and quality of the programs, their size and complexity will inevitably increase.

Hardware Requirements

The program was written and tested on a Commodore PET 2001-8K microcomputer. Since Commodore does not supply a printer yet, I entered the program with some modifications on an HP-3000 in order to get a program listing and printouts of the examples. It occupies about 7.5K bytes on the PET and twice as much on HP-3000. If you have less than 8K of available memory, delete all remarks to reduce the size of the program by about 1.5K. Further compression of the program can be achieved by leaving out the options you don't need.

When you expand your memory, you can add the missing modules or include additional calculations in the program. You are limited only by your memory capacity. All results are displayed on the standard output device, which is typically the CRT or hard-copy terminal, but if you desire, they could be directed to an optional printing device.

Human Aspects

All the prompts are self-explanatory, so anybody without computer knowledge can use the program without any prior training. Every time a new calculation is started, the screen is cleared in order to eliminate the scrolling of the previous lines, which can cause your vision to blur. This is not a problem on expensive terminals like HP-2645, but I recommend clearing the screen in all applications with less sophisticated CRTs or monitors. The user will surely be grateful to you.

Conclusion

In this structured program, several useful financial calculations have been gathered around one control module. As it is, it can be used by the layman or the professional, or it can be easily modified to accommodate more calculations, if desirable. In another instance, it can be compressed by deleting the options not desired. The program should also serve as an example of applying structured design and programming.■

Program listing. (Statement at line 2260 should be changed to IF X3 = (X2 – 1) THEN X$ = "0".)

```
10   REM   PROGRAM 'FINANC'.
11   REM   PROGRAMMED BY: LES PALENIK
12   REM        512-25 SILVERSPRINGS BLVD.
13   REM        SCARBORO,ONT.M1V 1M9,CANADA
14   REM
15   REM   THIS PROGRAM IS CURRENTLY BEING RUN
16   REM   ON COMMODORE PET-8K AND WITH SOME
17   REM   MODIFICATIONS ON HP-3000.
18   REM
30   REM   THE PROGRAM IS INTERACTIVE AND PROMPTS
31   REM   THE USER FOR ALL NECESSARY DATA.
32   REM   IT CONTAINS FOLLOWING APPLICATIONS:
33   REM   1. INVESTMENTS
34   REM    1.1. FUTURE VALUE OF ONE-TIME INVESTMENT
35   REM    1.2. FUTURE VALUE OF REGULAR DEPOSITS
36   REM    1.3. REQUIRED REGULAR DEPOSITS
37   REM
38   REM   2. DEPRECIATION
39   REM    2.1. DEPRECIATION RATE
40   REM    2.2. DEPRECIATION AMOUNT
```

```
41   REM   2.3. SALVAGE VALUE
42   REM
43   REM  3. LOANS
44   REM   3.1. REGULAR PAYMENT
45   REM   3.2. LAST PAYMENT
46   REM   3.3. TERM OF A LOAN
47   REM   3.4. 30  REMAINING BALANCE
48   REM   3.5. COST OF BORROWING
49   REM
50   REM
51   REM  INITIALIZE H$ WITH CODE TO CLEAR SCREEN
52   REM  AND L1$ WITH YOUR FAVORITE UNDERLINING
54    H$=' '
55    L1$='--------------------------------------------------'
100  PRINT H$
105  REM  ******************************
106  REM  MAIN-LINE     MODUL 0.
107  REM  ******************************
110  PRINT 'FINANCE PROGRAM'
115  PRINT '==============='
120  PRINT:PRINT
125  PRINT '1 - INVESTMENTS'
130  PRINT '2 - DEPRECIATION'
135  PRINT '3 - LOANS'
140  PRINT '0 - END OF PROGRAM'
145  REM  ******************************
146  REM  PROMPT FOR SELECTION
147  REM  ******************************
150  GOSUB 2300
155  REM  ******************************
156  REM  TEST ENTERED VALUE
157  REM  ******************************
160  IF A1=0 GOTO 195
165  IF A1<4 GOTO 185
170  REM  ******************************
171  REM  IF INVALID NO.=> ERROR MESSAGE
172  REM  ******************************
175  GOSUB 2000
180  GOTO 100
182  REM  ******************************
183  REM  ACTIVATE SELECTED GROUP
184  REM  ******************************
185  ON A1 GOSUB 300,400,500
190  GOTO 100
195  PRINT'*** END OF PROGRAM ***'
199  END

300  PRINT H$
305  REM  ******************************
306  REM    MODUL 1.
307  REM  ******************************
310  PRINT 'INVESTMENT CALCULATIONS'
315  PRINT '======================='
320  PRINT:PRINT
325  PRINT '1 - FUTURE VALUE OF ONE-TIME INVESTMENT'
330  PRINT '2 - FUTURE VALUE OF REGULAR DEPOSITS'
335  PRINT '3 - REGULAR DEPOSITS REQUIRED TO CREATE'
336  PRINT '     A DESIRED TOTAL VALUE'
340  PRINT '0 - END OF THIS GROUP'
345  REM  ******************************
346  REM  PROMPT FOR SELECTION
347  REM  ******************************
350  GOSUB 2300
355  REM  ******************************
356  REM  TEST ENTERED VALUE
357  REM  ******************************
360  IF A1=0 GOTO 399
365  IF A1<4 GOTO 385
370  REM  ******************************
371  REM  IF INVALID NO.=> ERROR MESSAGE
372  REM  ******************************
375  GOSUB 2000
380  GOTO 300
382  REM  ******************************
383  REM  ACTIVATE SELECTED OPTION
384  REM  ******************************
385  ON A1 GOSUB 600,700,800
390  GOTO 300
399  RETURN

400  PRINT H$
405  REM  ******************************
406  REM    MODUL 2.
407  REM  ******************************
410  PRINT 'DEPRECIATION CALCULATIONS'
415  PRINT '========================='
420  PRINT:PRINT
425  PRINT '1 - ANNUAL DEPRECIATION RATE'
430  PRINT '2 - AMOUNT OF DEPRECIATION'
435  PRINT '3 - SALVAGE VALUE'
440  PRINT '0 - END OF THIS GROUP'
445  REM  ******************************
446  REM  PROMPT FOR SELECTION
447  REM  ******************************
450  GOSUB 2300
455  REM  ******************************
456  REM  TEST ENTERED VALUE
457  REM  ******************************
460  IF A1=0 GOTO 499
465  IF A1<4 GOTO 485
470  REM  ******************************
471  REM  IF INVALID NO.=> ERROR MESSAGE
472  REM  ******************************
475  GOSUB 2000
480  GOTO 400
482  REM  ******************************
483  REM  ACTIVATE SELECTED OPTION
484  REM  ******************************
485  ON A1 GOSUB 1000,1100,1200
490  GOTO 400
499  RETURN
500  PRINT H$
505  REM  ******************************
506  REM    MODUL 3.
507  REM  ******************************
510  PRINT 'LOAN CALCULATIONS'
515  PRINT '================='
520  PRINT:PRINT
525  PRINT '1 - REGULAR PAYMENT ON A LOAN'
530  PRINT '2 - LAST PAYMENT ON A LOAN'
535  PRINT '3 - TERM OF A LOAN'
536  PRINT '4 - REMAINING BALANCE ON A LOAN'
537  PRINT '5 - COST OF BORROWING'
540  PRINT '0 - END OF THIS GROUP'
545  REM  ******************************
546  REM  PROMPT FOR SELECTION
547  REM  ******************************
550  GOSUB 2300
555  REM  ******************************
556  REM  TEST ENTERED VALUE
557  REM  ******************************
560  IF A1=0 GOTO 599
565  IF A1<6 GOTO 585
570  REM  ******************************
571  REM  IF INVALID NO.=> ERROR MESSAGE
572  REM  ******************************
575  GOSUB 2000
580  GOTO 500
582  REM  ******************************
583  REM  ACTIVATE SELECTED OPTION
584  REM  ******************************
585  ON A1 GOSUB 1500,1600,1700,1800,1900
590  GOTO 500
599  RETURN

600  PRINT H$
605  REM  ******************************
606  REM    MODUL 1.1.
607  REM  ******************************
610  PRINT 'FUTURE VALUE OF ONE-TIME INVESTMENT'
615  PRINT '==================================='
620  PRINT:PRINT
625  PRINT 'ENTER INITIAL INVESTMENT            :';
626  INPUT I1
630  PRINT 'ENTER NOMINAL INTEREST RATE         :';
631  INPUT R
635  PRINT 'TERM OF INVESTMENT(YEARS,MONTHS)    :';
636  INPUT Y,M
640  PRINT 'NUMBER OF COMPOUNDING PERIODS / YEAR:';
641  INPUT N
645  REM  ******************************
646  REM  CALCULATIONS
647  REM  ******************************
650  R=R/N/100
655  Y=(12*Y+M)/12
660  T2=I1*(1+R)↑(N*Y)
665  T2=INT(T2*100+.5)/100
670  I2=T2-I1
675  REM  ******************************
676  REM  DISPLAY RESULTS AND ASK IF
677  REM  ANOTHER CALCULATION IS DESIRED
678  REM  ******************************
690  GOSUB 2100
692  GOSUB 2050
695  IF A$='Y' GOTO 600
699  RETURN
700  PRINT H$
705  REM  ******************************
706  REM    MODUL 1.2.
707  REM  ******************************
710  PRINT 'FUTURE VALUE OF REGULAR DEPOSITS'
715  PRINT '================================'
720  PRINT:PRINT
725  PRINT 'ENTER AMOUNT OF EACH DEPOSIT        :';
726  INPUT D
730  PRINT 'ENTER NOMINAL INTEREST RATE         :';
731  INPUT R
735  PRINT 'TERM OF INVESTMENT(YEARS,MONTHS)    :';
736  INPUT Y,M
740  PRINT 'NUMBER OF DEPOSITS PER YEAR         :';
741  INPUT N
745  REM  ******************************
746  REM  CALCULATIONS
747  REM  ******************************
750  R=R/N/100
755  Y=(12*Y+M)/12
760  T2=D*((1+R)↑(N*Y)-1)/R
765  T2=INT(T2*100+.5)/100
770  I1=D*Y*N
775  I2=T2-I1
```

```
780  REM  ******************************
785  REM  DISPLAY RESULTS AND ASK IF
787  REM  ANOTHER CALCULATION IS DESIRED
788  REM  ******************************
790  GOSUB 2100
792  GOSUB 2050
795  IF A$='Y' GOTO 700
799  RETURN

800  PRINT H$
805  REM  ******************************
806  REM    MODUL 1.3.
807  REM  ******************************
810  PRINT 'REQUIRED REGULAR DEPOSITS'
815  PRINT '========================='
820  PRINT:PRINT
825  PRINT 'ENTER THE DESIRED TOTAL VALUE       :';
826  INPUT T2
830  PRINT 'ENTER NOMINAL INTEREST RATE         :';
831  INPUT R
835  PRINT 'TERM OF INVESTMENT(YEARS,MONTHS)    :';
836  INPUT Y,M
840  PRINT 'NUMBER OF DEPOSITS PER YEAR         :';
841  INPUT N
845  REM  ******************************
846  REM  CALCULATIONS
847  REM  ******************************
850  R=R/N/100
855  Y=(12*Y+M)/12
860  D=T2*R/((1+R)↑(N*Y)-1)
865  D=INT(D*100+.5)/100
866  X1=D:GOSUB 2200
870  I1=D*Y*N
875  I2=T2-I1
880  REM  ******************************
885  REM  DISPLAY RESULTS AND ASK IF
887  REM  ANOTHER CALCULATION IS DESIRED
888  REM  ******************************
889  PRINT:PRINT
890  PRINT 'REGULAR DEPOSITS:';X$
891  GOSUB 2100
892  GOSUB 2050
895  IF A$='Y' GOTO 800
899  RETURN

1000 PRINT H$
1005 REM  ******************************
1006 REM    MODUL 2.1.
1007 REM  ******************************
1010 PRINT 'ANNUAL DEPRECIATION RATE'
1015 PRINT '========================'
1020 PRINT:PRINT
1025 PRINT 'ENTER ORIGINAL VALUE                :';
1026 INPUT P1
1030 PRINT 'ENTER THE RESALE VALUE              :';
1031 INPUT P2
1035 PRINT 'DEPRECIATION TERM (YEARS,MONTHS)    :';
1036 INPUT Y,M
1045 REM  ******************************
1046 REM  CALCULATIONS
1047 REM  ******************************
1050 Y=(12*Y+M)/12
1055 R=1-(P2/P1)↑(1/Y)
1060 R=INT(R*100+.05)
1065 REM  ******************************
1070 REM  DISPLAY RESULTS AND ASK IF
1071 REM  ANOTHER CALCULATION IS DESIRED
1072 REM  ******************************
1075 PRINT:PRINT
1080 PRINT 'DEPRECIATION RATE:',R;'%'
1082 PRINT L1$
1085 GOSUB 2050
1095 IF A$='Y' GOTO 1000
1099 RETURN

1100 PRINT H$
1105 REM  ******************************
1106 REM    MODUL 2.2.
1107 REM  ******************************
1110 PRINT 'DEPRECIATION AMOUNT'
1115 PRINT '==================='
1120 PRINT:PRINT
1125 PRINT 'ENTER THE ORIGINAL VALUE            :';
1126 INPUT P1
1130 PRINT 'ENTER THE DEPRECIATION RATE         :';
1131 INPUT R
1135 PRINT 'ENTER THE YEAR OF DEPRECIATION      :';
1136 INPUT Y
1140 REM  ******************************
1141 REM  CALCULATIONS
1142 REM  ******************************
1150 R=R/100
1155 P3=P1*R*(1-R)↑(Y-1)
1160 X1=P3
1165 GOSUB 2200
1170 REM  ******************************
1171 REM  DISPLAY RESULTS AND ASK IF
1172 REM  ANOTHER CALCULATION IS DESIRED
1173 REM  ******************************
1175 PRINT:PRINT
```

```
1180 PRINT 'AMOUNT OF DEPRECIATION:';X$
1182 PRINT L1$
1190 GOSUB 2050
1195 IF A$='Y' GOTO 1100
1199 RETURN

1200 PRINT H$
1205 REM  ******************************
1206 REM    MODUL 2.3.
1207 REM  ******************************
1210 PRINT 'SALVAGE VALUE'
1215 PRINT '============='
1220 PRINT:PRINT
1225 PRINT 'ENTER THE ORIGINAL VALUE            :';
1226 INPUT P1
1230 PRINT 'ENTER THE DEPRECIATION RATE         :';
1231 INPUT R
1235 PRINT 'DEPRECIATION TERM (YEARS,MONTHS)    :';
1236 INPUT Y,M
1245 REM  ******************************
1246 REM  CALCULATIONS
1247 REM  ******************************
1250 R=R/100
1255 S=P1*(1-R)↑Y
1260 X1=S
1265 GOSUB 2200
1270 REM  ******************************
1271 REM  DISPLAY RESULTS AND ASK IF
1272 REM  ANOTHER CALCULATION IS DESIRED
1273 REM  ******************************
1275 PRINT:PRINT
1280 PRINT 'SALVAGE VALUE:';X$
1282 PRINT L1$
1285 GOSUB 2050
1295 IF A$='Y' GOTO 1200
1299 RETURN

1500 PRINT H$
1505 REM  ******************************
1506 REM    MODUL 3.1.
1507 REM  ******************************
1510 PRINT 'REGULAR PAYMENT ON A LOAN'
1515 PRINT '========================='
1520 PRINT:PRINT
1525 PRINT 'ENTER THE PRINCIPAL AMOUNT          :';
1526 INPUT P1
1530 PRINT 'ENTER TERM OF LOAN (YEARS,MONTHS)   :';
1531 INPUT Y,M
1535 PRINT 'ENTER THE ANNUAL INTEREST RATE      :';
1536 INPUT R
1540 PRINT 'NUMBER OF PAYMENTS PER YEAR         :';
1541 INPUT N
1545 REM  ******************************
1546 REM  CALCULATIONS
1547 REM  ******************************
1550 R=R/N/100:Y=(Y*12+M)/12
1555 P3=1/(1+R)↑(N*Y)
1560 P2=P1*R/(1-P3)
1565 X1=P2:GOSUB 2200
1570 REM  ******************************
1571 REM  DISPLAY RESULTS AND ASK IF
1572 REM  ANOTHER CALCULATION IS DESIRED
1573 REM  ******************************
1580 PRINT:PRINT
1585 PRINT 'REGULAR PAYMENT:';X$
1587 PRINT L1$
1590 GOSUB 2050
1595 IF A$='Y' GOTO 1500
1599 RETURN

1600 PRINT H$
1605 REM  ******************************
1606 REM    MODUL 3.2.
1607 REM  ******************************
1610 PRINT 'LAST PAYMENT ON A LOAN'
1615 PRINT '======================'
1620 PRINT:PRINT
1625 PRINT 'ENTER THE PRINCIPAL AMOUNT          :';
1626 INPUT P1
1630 PRINT 'ENTER TERM OF LOAN (YEARS,MONTHS)   :';
1631 INPUT Y,M
1635 PRINT 'ENTER THE ANNUAL INTEREST RATE      :';
1636 INPUT R
1640 PRINT 'NUMBER OF PAYMENTS PER YEAR         :';
1641 INPUT N
1643 PRINT 'ENTER AMOUNT OF REGULAR PAYMENT     :';
1644 INPUT P4
1645 REM  ******************************
1646 REM  CALCULATIONS
1647 REM  ******************************
1650 R=R/N/100:Y=(Y*12+M)/12
1652 I1=N*Y
1654 FOR I=1 TO I1
1656 R1=INT(P1*R*100+.5)/100
1658 R2=P4-R1
1660 P1=P1-R2
1662 NEXT I
1664 P2=P4+P1
1666 X1=P2:GOSUB 2200
1670 REM  ******************************
```

```
1671 REM  DISPLAY RESULTS AND ASK IF
1672 REM  ANOTHER CALCULATION IS DESIRED
1673 REM  ******************************
1677 PRINT:PRINT
1680 PRINT 'LAST PAYMENT:';X$
1682 PRINT L1$
1685 GOSUB 2050
1695 IF A$='Y' GOTO 1600
1699 RETURN

1700 PRINT H$
1705 REM  ******************************
1706 REM    MODUL 3.3.
1707 REM  ******************************
1710 PRINT 'TERM OF A LOAN'
1715 PRINT '=============='
1720 PRINT:PRINT
1725 PRINT 'ENTER THE PRINCIPAL AMOUNT          :';
1726 INPUT P1
1730 PRINT 'ENTER AMOUNT OF REGULAR PAYMENT     :';
1731 INPUT P4
1735 PRINT 'ENTER THE ANNUAL INTEREST RATE      :';
1736 INPUT R
1740 PRINT 'NUMBER OF PAYMENTS PER YEAR         :';
1741 INPUT N
1745 REM  ******************************
1746 REM  CALCULATIONS
1747 REM  ******************************
1750 R=R/N/100
1752 T1=1-(P1*R/P4)
1754 T2=1+R
1756 T=-(LOG(T1)/LOG(T2))/N
1758 M=INT(T*12)
1760 Y=INT(M/12)
1762 M=M-Y*12
1765 REM  ******************************
1770 REM  DISPLAY RESULTS AND ASK IF
1771 REM  ANOTHER CALCULATION IS DESIRED
1772 REM  ******************************
1775 PRINT:PRINT
1780 PRINT 'TERM OF LOAN:';Y;'YEARS,';M;'MONTHS'
1782 PRINT L1$
1785 GOSUB 2050
1795 IF A$='Y' GOTO 1700
1799 RETURN

1800 PRINT H$
1805 REM  ******************************
1806 REM    MODUL 3.4.
1807 REM  ******************************
1810 PRINT 'REMAINING BALANCE ON A LOAN'
1815 PRINT '==========================='
1820 PRINT:PRINT
1825 PRINT 'ENTER THE PRINCIPAL AMOUNT          :';
1826 INPUT P1
1830 PRINT 'ENTER AMOUNT OF REGULAR PAYMENT     :';
1831 INPUT P4
1835 PRINT 'ENTER THE ANNUAL INTEREST RATE      :';
1836 INPUT R
1840 PRINT 'NUMBER OF PAYMENTS PER YEAR         :';
1841 INPUT N
1842 PRINT 'NUMBER OF PAYMENTS MADE             :';
1843 INPUT I1
1845 REM  ******************************
1846 REM  CALCULATIONS
1847 REM  ******************************
1850 R=R/N/100
1852 FOR I=1 TO I1
1854 R1=INT(P1*R*100+.5)/100
1856 R2=P4-R1
1858 P1=P1-R2
1860 NEXT I
1862 X1=P1:GOSUB 2200
1865 REM  ******************************
1870 REM  DISPLAY RESULTS AND ASK IF
1871 REM  ANOTHER CALCULATION IS DESIRED
1872 REM  ******************************
1875 PRINT:PRINT
1880 PRINT 'REMAINING BALANCE:';X$
1882 PRINT L1$
1885 GOSUB 2050
1895 IF A$='Y' GOTO 1800
1899 RETURN

1900 PRINT H$
1905 REM  ******************************
1906 REM    MODUL 3.5.
1907 REM  ******************************
1910 PRINT 'COST OF BORROWING'
1915 PRINT '================='
1920 PRINT:PRINT
1925 PRINT 'ENTER THE PRINCIPAL AMOUNT          :';
1926 INPUT P1
1930 PRINT 'ENTER TERM OF LOAN (YEARS,MONTHS)   :';
1931 INPUT Y,M
1935 PRINT 'ENTER THE ANNUAL INTEREST RATE      :';
1936 INPUT R
1940 PRINT 'NUMBER OF PAYMENTS PER YEAR         :';
1941 INPUT N
1945 REM  ******************************
1946 REM  CALCULATIONS
1947 REM  ******************************
1950 R=R/N/100:Y=(Y*12+M)/12
1955 P3=1/(1+R)↑(N*Y)
1960 P4=P1*R/(1-P3)
1962 P4=INT(P4*100+.5)/100
1964 P5=P1:C=0
1966 I1=N*Y
1968 FOR I=1 TO I1
1970 R1=INT(P5*R*100+.5)/100
1972 R2=P4-R1:P5=P5-R2
1974 C=C+P4
1976 NEXT I
1978 C=C+P5:C1=C-P1
1980 REM  ******************************
1981 REM  DISPLAY RESULTS AND ASK IF
1982 REM  ANOTHER CALCULATION IS DESIRED
1983 REM  ******************************
1984 PRINT:PRINT L1$:PRINT
1985 X1=P4:GOSUB 2200
1986 PRINT 'REGULAR PAYMENT:';X$
1987 X1=C:GOSUB 2200
1988 PRINT 'TOTAL PAYMENTS:';X$
1989 X1=C1:GOSUB 2200
1990 PRINT 'COST OF BORROWING:';X$
1991 PRINT :PRINT L1$
1992 GOSUB 2050
1995 IF A$='Y' GOTO 1900
1999 RETURN

2000 REM  ******************************
2001 REM  ERROR MESSAGE SUBROUTINE
2002 REM  ******************************
2005 PRINT
2010 PRINT '**** INVALID DATA     RETRY ****'
2015 PRINT
2020 FOR I=1 TO 500
2025 A1=ABS(A1)
2030 NEXT I
2035 RETURN

2050 REM  ******************************
2051 REM  THIS SUBROUTINE WILL ASK IF
2052 REM  MORE CALCULATIONS FOR THE SPE-
2053 REM  CIFIC OPTION ARE DESIRED.
2054 REM  ******************************
2060 PRINT
2065 PRINT 'ANOTHER CALCULATION (Y OR N) ';
2070 INPUT A$
2075 IF (A$='Y') OR (A$='N') GOTO 2095
2080 GOSUB 2000
2085 GOTO 2060
2095 PRINT
2099 RETURN

2100 REM  ******************************
2101 REM  THIS SUBROUTINE WILL FORMAT &
2102 REM  DISPLAY: INVESTMENT VALUE,
2103 REM  INTEREST AND TOTAL VALUE
2104 REM  ******************************
2105 PRINT
2110 PRINT L1$
2115 PRINT
2120 X1=I1:GOSUB 2200
2125 PRINT 'VALUE OF YOUR INVESTMENT:';X$
2130 X1=I2:GOSUB 2200
2135 PRINT 'VALUE OF ACCUM.INTEREST :';X$
2140 X1=T2:GOSUB 2200
2145 PRINT 'TOTAL VALUE             :';X$
2150 PRINT L1$
2155 PRINT
2160 RETURN

2200 REM ******************************
2201 REM  THIS SUBROUTINE WILL FORMAT
2202 REM  DOLLARS AND CENTS
2203 REM  ******************************
2205 X1=INT(X1*100+.5)/100
2210 X$=''
2215 X1$=STR$(X1)
2220 X2=LEN(X1$)
2225 FOR I=1 TO X2
2230 X2$=MID$(X1$,I,1)
2235 X3=I
2240 IF X2$='.' GOTO 2260
2245 NEXT I
2250 X$='.00'
2255 GOTO 2265
2260 IF X2=X3 THEN X$='0'
2265 X$='$'+X1$+X$
2270 RETURN

2300 REM  ******************************
2301 REM  THIS SUBROUTINE PROMPTS FOR
2302 REM  SELECTION OF AN OPTION
2303 REM  ******************************
2305 PRINT:PRINT
2310 PRINT 'SELECT ONE OF THE OPTIONS ABOVE ';
2315 INPUT A1
2320 RETURN
```

Computer-Generated Signs: Put Your TTY to Work!

The ability to generate signs, for unlimited applications, is a useful feature on any system. Here's a short, and simple, BASIC program for this fun application.

Joseph J. Roehrig
Box 74
Middle Village NY 11379

Some people who get involved with microprocessors have never programmed a computer before. The most common language used by the beginner is BASIC, and a lot of instructional material is available for it. However, most of the available material does not go into the step-by-step logic needed to create a program. The purpose of this article is to describe the step-by-step construction of a program that will print signs on a single sheet of 14 7/8 x 11 inch computer paper. This paper is 132 characters wide and 66 characters high. For our signs, there will be 7 lines and 10 letters per line. Between each 10 x 7 letter, two blanks will be left, making each letter 12 x 9. This will fit because 12 (letter width) x 10 (letters per line) = 120, which is less than 132 (paper size width) and 9 (letter width) x 7 (number of lines) = 63, which is less than 66 (paper size height).

How the Program Works

Suppose we want to make a sign seven lines long with ten letters per line. Let's start to write the program. First, how many different letters do we need? To keep the problem relatively short, we will choose 27 letters (26 for the alphabet plus one for a blank). Later, when you understand the program completely, you can add all the numbers and special characters desired.

A string variable, a variable that corresponds to a character rather than a numeric value, will be needed for each letter. The letter A is a string variable, as well as the number 1. Numbers can be either string variables or numeric variables, while letters and special characters can only be string variables. The easy way to manipulate 27 variables is by using a matrix or array because you can refer to different variables by the same name and the particular index number, rather than by 27 different names. The variable L$(27) is selected and dimensioned to 27 positions. The $ indicates that this is a string variable rather than numeric. In order to have 27 positions, it must be dimensioned in a DIM statement. The first line of the program is 10 DIM

Program listing.

```
10 DIMV(31),L$(27),L(27,7),Z(5),D$(70)
15 L$="ABCDEFGHIJKLMNOPQRSTUVWXYZ "
20 FORA=1TO5\READZ(A)\NEXT
25 D1$="          "
30 DATA10000,1000,100,10,1
40 FORA=1TO7\D$=D$+D1$\NEXT
50 FORA=0TO15\READV(A)\NEXT
52 DATA 0,1,10,11,100,101,110,111,1000,1001,1010,1011,1100
54 DATA 1101,1110,1111
60 FORA=16TO31\V(A)=10000+V(A-16)\NEXT
70 FORA=1TO27
80 FORB=1TO7\READL(A,B)\NEXT\NEXT
84 !"INPUT FULL LINES ONLY"\!" ----------"
90 FORA=1TO7\B=A*10\INPUTD1$\D$(B-9,B)=D1$\NEXT
98 INPUT"GET YOUR PAPER READY AND ENTER A BLANK",U$
100 FORA=1TO7
105 FORD=1TO7
110 FORB=1TO10
120 C=((A-1)*10)+B
122 C$=D$(C,C)\C=ASC(C$)
124 C=C-64\IF0>CTHENC=27
135 F=L(C,D)
136 F=V(F)
137 Q=C
150 FORE=1TO5
160 G=INT(F/Z(E))
165 F=F-(G*Z(E))
170 IFG=1THEN190
180 !"  ",\GOTO200
190 !L$(Q,Q),L$(Q,Q),
200 NEXTE\!"  ",
220 NEXTB\!" "
```

L$(27).

We want each position in L$ to represent a different letter and, finally, a blank. Line 15 accomplishes this.

How are we going to form our letters? Each letter will be the original 5 x 7 character configuration, with the 5 being doubled. We'll work with the undoubled 5, though, because it is smaller than 10 and, therefore, easier

```
EEEEEEEEEE
EE
EE
EEEEEEEE
EE
EE
EEEEEEEEEE
```

(Actual size.)

```
230 NEXTD\!" "\!" "
240 NEXTA\END
301 DATA 04,10,17,31,17,17,17, 30,17,17,31,17,17,30
302 DATA 14,17,16,16,16,17,14, 28,18,17,17,17,18,28
303 DATA 31,16,16,30,16,16,31, 31,16,16,30,16,16,16
304 DATA 14,17,16,16,19,17,14, 17,17,17,31,17,17,17
305 DATA 31, 4, 4, 4, 4, 4,31,  1, 1, 1, 1,17, 9, 6
306 DATA 17,18,20,24,20,18,17, 16,16,16,16,16,16,31
307 DATA 17,27,21,17,17,17,17, 17,25,21,21,21,19,17
308 DATA  4,10,17,17,17,10, 4, 30,17,17,30,16,16,16
309 DATA  4,10,17,17,21,10, 5, 30,17,17,30,20,18,17
310 DATA 14,17,16,14, 1,17,14, 31, 4, 4, 4, 4, 4, 4
311 DATA 17,17,17,17,17,17,14, 17,17,17,17,17,10, 4
312 DATA 17,17,17,17,21,31,17, 17,10,10,4,10,10,17
313 DATA 17,17,10,4,4,4,4,31,1,2,4,4,8,31,0,0,0,0,0,0,0
READY
RUN

INPUT FULL LINES ONLY
---------------------
?ABCDEFGHIJ
?KLMNOPQRST
?UVWXYZ
?   SKIP
?THIS IS A
?SAMPLE OF
?CHARACTERS
GET YOUR PAPER READY AND ENTER A BLANK
```

```
    AA     BBBBBBBB     CCCCCC   DDDDDD     EEEEEEEEEE FFFFFFFFFF   GGGGGG   HH      HH IIIIIIIIII         JJ
  AA  AA   BB      BB CC      CC DD    DD   EE         FF         GG      GG HH      HH     II             JJ
AA      AA BB      BB CC         DD      DD EE         FF         GG         HH      HH     II             JJ
AAAAAAAAAA BBBBBBBBBB CC         DD      DD EEEEEEEE   FFFFFFFF   GG         HHHHHHHHHH     II             JJ
AA      AA BB      BB CC         DD      DD EE         FF         GG    GGGG HH      HH     II     JJ      JJ
AA      AA BB      BB CC      CC DD    DD   EE         FF         GG      GG HH      HH     II       JJ    JJ
AA      AA BBBBBBBB     CCCCCC   DDDDDD     EEEEEEEEEE FF           GGGGGG   HH      HH IIIIIIIIII     JJJJ

KK      KK LL         MM      MM NN      NN     OO     PPPPPPPP       QQ     RRRRRRRR     SSSSSS   TTTTTTTTTT
KK    KK   LL         MMMM  MMMM NNNN    NN   OO  OO   PP      PP   QQ  QQ   RR      RR SS      SS     TT
KK  KK     LL         MM  MM  MM NN  NN  NN OO      OO PP      PP QQ      QQ RR      RR SS             TT
KKKK       LL         MM      MM NN  NN  NN OO      OO PPPPPPPP   QQ      QQ RRRRRRRR     SSSSSS       TT
KK  KK     LL         MM      MM NN  NN  NN OO      OO PP         QQ  QQ  QQ RR  RR             SS     TT
KK    KK   LL         MM      MM NN    NNNN   OO  OO   PP           QQ  QQ   RR    RR   SS      SS     TT
KK      KK LLLLLLLLLL MM      MM NN      NN     OO     PP             QQ  QQ RR      RR   SSSSSS       TT

UU      UU VV      VV WW      WW XX      XX YY      YY ZZZZZZZZZZ
UU      UU VV      VV WW      WW   XX  XX   YY      YY         ZZ
UU      UU VV      VV WW      WW   XX  XX     YY  YY         ZZ
UU      UU VV      VV WW      WW     XX         YY         ZZ
UU      UU VV      VV WW  WW  WW   XX  XX       YY         ZZ
UU      UU   VV  VV   WWWWWWWWWW   XX  XX       YY       ZZ
  UUUUUU       VV     WW      WW XX      XX     YY     ZZZZZZZZZZ

                                   SSSSSS   KK      KK IIIIIIIIII PPPPPPPP
                                 SS      SS KK    KK       II     PP      PP
                                 SS         KK  KK         II     PP      PP
                                   SSSSSS   KKKK           II     PPPPPPPP
                                         SS KK  KK         II     PP
                                 SS      SS KK    KK       II     PP
                                   SSSSSS   KK      KK IIIIIIIIII PP

TTTTTTTTTT HH      HH IIIIIIIIII   SSSSSS              IIIIIIIIII   SSSSSS                  AA
    TT     HH      HH     II     SS      SS                II     SS      SS              AA  AA
    TT     HH      HH     II     SS                        II     SS                    AA      AA
    TT     HHHHHHHHHH     II       SSSSSS                  II       SSSSSS              AAAAAAAAAA
    TT     HH      HH     II             SS                II             SS            AA      AA
    TT     HH      HH     II     SS      SS                II     SS      SS            AA      AA
    TT     HH      HH IIIIIIIIII   SSSSSS              IIIIIIIIII   SSSSSS              AA      AA

  SSSSSS       AA     MM      MM PPPPPPPP   LL         EEEEEEEEEE                OO     FFFFFFFFFF
SS      SS   AA  AA   MMMM  MMMM PP      PP LL         EE                      OO  OO   FF
SS         AA      AA MM  MM  MM PP      PP LL         EE                    OO      OO FF
  SSSSSS   AAAAAAAAAA MM      MM PPPPPPPP   LL         EEEEEEEE              OO      OO FFFFFFFF
        SS AA      AA MM      MM PP         LL         EE                    OO      OO FF
SS      SS AA      AA MM      MM PP         LL         EE                      OO  OO   FF
  SSSSSS   AA      AA MM      MM PP         LLLLLLLLLL EEEEEEEEEE                OO     FF

  CCCCCC   HH      HH     AA     RRRRRRRR       AA       CCCCCC   TTTTTTTTTT EEEEEEEEEE RRRRRRRR     SSSSSS
CC      CC HH      HH   AA  AA   RR      RR   AA  AA   CC      CC     TT     EE         RR      RR SS      SS
CC         HH      HH AA      AA RR      RR AA      AA CC             TT     EE         RR      RR SS
CC         HHHHHHHHHH AAAAAAAAAA RRRRRRRR   AAAAAAAAAA CC             TT     EEEEEEEE   RRRRRRRR     SSSSSS
CC         HH      HH AA      AA RR  RR     AA      AA CC             TT     EE         RR  RR             SS
CC      CC HH      HH AA      AA RR    RR   AA      AA CC      CC     TT     EE         RR    RR   SS      SS
  CCCCCC   HH      HH AA      AA RR      RR AA      AA   CCCCCC       TT     EEEEEEEEEE RR      RR   SSSSSS
```

Loop	Starts at Statement	Ends at Statement	Purpose
1	100	240	Counts the seven lines of the message.
2	105	230	Counts the seven lines within each letter.
3	110	220	Counts the ten letters in each line.
4	150	200	Counts the five characters composing each letter.

Fig. 2.

to use. A test letter, H, can be described as follows:

10001
10001
10001
11111
10001
10001
10001

As we can see, the letter is seven lines of 5 digit patterns (binary digits). The maximum number of patterns is 2^5, or 32. You can prove this by writing all of the different five-digit numbers possible using only 0s and 1s (the lowest is 00000 and the highest is 11111). These 32 patterns can be used to form any possible combination; however, we must tell the computer what these patterns are.

Once again, a variable will be dimensioned for this purpose, and V(31) is added to statement 10. Thirty-one is being used (not 32) because V is a numeric rather than a string variable; and with the BASIC being used (North Star) the 0 index of a numeric variable is legal. Line 50 is used to read in the patterns for V(0) to V(15), and lines 52 and 54 contain the first 16 patterns. The remaining 16 patterns merely add 10000 to the first 16. This is done in line 60 by indexing. The variable A is set to loop from 16 to 31. Therefore, on the first pass:

V(A)=10000+V(A-16)
or
V(16)=10000+V(16-16)
or
V(16)=10000+V(0)
or
V(16)=10000+0
or
V(16)=10000

This is repeated until V(31)=10000+V(15) or V(31)=10000+1111. All of the patterns are now set.

The next step is to assign the seven patterns to each of the letters required to print the seven lines of each letter. Another variable is added to line 10 – a two-dimensional variable, L(27,7). In a two-dimensional variable each of the 27 variables created by the first dimension now has seven subdivisions. The new variable, L(27,7), is closely related (logically not in computer code) to variable L$(27). Even though L(27,7) is numeric, we'll forget about the 0 dimensions in order to be able to use the same index numbers as L$(27). Each of the 27 first dimensions of L(27,7) has seven subdivisions that correspond to the seven lines of each letter. In order to link it to L$(27), L(1,7) will represent A, L(8,7) represents H, etc.

Lines 70 and 80 read the 27 patterns of seven digits. The outer loop is in line 70 and has A as the index. Line 80 has the B index and provides the inner loop. Both loops end in line 80. Therefore, for each A from 1 to 27 that is indexed, seven Bs from one to seven are read before the A index is incremented to the next number. The data read is contained in sequence from line 301 to 313.

A simple message, "Input Full Lines Only," is printed in statement 84.

Before a message can be input, variable D$(70) must be set to blanks for each of its 70 characters. Statement 25 sets D1$ equal to ten blanks. In line 40, D$ is set equal to D1$ seven times, or for 7 x 10 blanks.

The seven lines of the sign are entered in line 90. The line index is A and is set for 1 to 7 corresponding to the lines. The variable B is set to the last character in the line by having B equal 10*A. A total of 70 characters comprises the message in Fig. 1.

Line 1	Characters 1-10
Line 2	Characters 11-20
Line 3	Characters 21-30
Line 4	Characters 31-40
Line 5	Characters 41-50
Line 6	Characters 51-60
Line 7	Characters 61-70

Fig. 1.

First, each of the seven inputs is placed in a temporary variable, D1$. This data is then transferred to the main D$ variable by saying that D1$ will represent the B-9 to B character of variable D$. As mentioned, B will always equal the last character of a line. Therefore, B-9 must equal the first character. The reason for using D1$ is to allow the user to input his message line by line rather than asking for the entire string of 70 characters at one time.

A simple message is printed by the Input statement contained in line 98. The string variable U$ is read; after A the message is printed. Although U$ is not used, it is in the program as a timing delay. The user can adjust his paper and start the sign printing anytime he desires because of this delay-type statement.

Before we can print a message, we must be able to break down the five-digit V(31) codes into five individual digits. The variable Z(5) is added to line 10, line 20 reads in the five values, and the data for Z is contained in statement number 30. The Z variable was set to:

Z(1) = 10000
Z(2) = 1000
Z(3) = 100
Z(4) = 10
Z(5) = 1

These variables will be used to break down the V codes.

We're ready to print. The printing will be done by using four loops. Fig. 2 contains a description of the four loops, with loop 1 being the outermost and loop 4 being the innermost.

In addition to ending the loop: line 200 places two blanks between each character; line 220 changes to the next print line; line 230 prints two blank lines between each of the seven lines of the message; and line 240 ends the program. The tasks in Fig. 3 are accomplished between lines 120 and 137.

We now encounter line 150, which starts the innermost loop. This loop prints the actual characters of each letter. As in our sample, the letter H is a 17, 17, 17, 31, 17, 17, 17 configuration representing the seven lines

Line	Purpose
120	Sets variable C to the character of the message currently being printed. A is the line number; thus (A-1)*10 equals the letters already printed on previous lines. B represents the letter of the current line being printed.
122	Sets C$ to the current letter being printed, since D$ is the total message and C is the current letter. C is then reset to correspond to the ASCII code of the actual letter in the message. This is done by using the BASIC function ASC.
124	Since A=1 in our program, B=2, etc., and the ASCII code for A=65, B=66, etc.; 64 is taken away from C. All other characters not between A and Z are set to our code of 27 or blank.
135	Sets F equal to L(C,D) with C equaling the letter and D corresponding to the line of the letter between 1 and 7. (As mentioned earlier, each letter is ten characters wide and seven lines tall.) Variable L(C,D) equals the pattern to be printed.
136	Sets F to one of V(F) 32 patterns.
137	Sets Q equal to C.

Fig. 3.

needed to form an H. The first 17 corresponds to 10001. So far, lines 120 to 137 have done the following (assuming H is the first letter of the first line in our message):

Line	Function Accomplished
120	C=1
122	C$ = H and C = 72
124	C=8
135	F=L(8,1) F=17
136	F=V(F) F=10001
137	Q=8

Fig. 4 shows what the loop started in line 150 does in lines 160 and 165. This illustrates the breaking of the code into five separate elements. In line 170, if G equals 1, we branch to 190, the print statement. Here the letter of the message is printed twice in order to increase our letter width from five characters to ten (five passes of two characters).

If G had equaled zero, line 180 would have printed two blanks. Both lines 180 and 190 go to line 200 after printing. Therefore, our sample letter, H, will be printed twice, then six blanks, and another two Hs. The program keeps looping until the message is complete.

Pass	Value of G
	G=INT(F/Z(E)):F=F-(G*Z(E))
1	1=INT(10001/10000):1=10001-(1*10000)
2	0=INT(1/1000):1=1-(0*1000)
3	0=INT(1/100):1=1-(0*100)
4	0=INT(1/10):1=1-(0*10)
5	1=INT(1/1):0=1-(1*1)

Fig. 4.

Auld Lang Signs

From the above description, you should now be able to expand the program to print numbers and special characters. Here are some hints, if you need them.

1. Expand L$(27) to include all of the characters desired.
2. Change line 15 to include the added characters.
3. Expand L(27,7) to include the added characters.
4. Add data after line 313 for the new characters — seven patterns per character.
5. Change line 70 to equal the number of characters.
6. Make sure the C=ASC (C$) in lines 122 and 124 equals one of your letters. ■

Copying Computer Cassettes

Constant use of cassette tapes can damage them. As a safeguard, copies should be made; this gadget makes duplication easy.

Peter A. Stark
PO Box 209
Mt. Kisco NY 10549

No, I'm *not* suggesting that you acquire cheap software by copying somebody else's cassettes. On the other hand, it is a good idea to make duplicate copies of your own tapes to protect the originals from possible damage caused by constant use.

When I purchased my SWTP system, I bought the usual collection of programs on Kansas City cassettes—4K and 8K BASIC, an editor and assembler, a disassembler and some games. Later, I purchased an extremely sophisticated text editor and text processor package from Technical Systems Consultants. Before I knew it, I had a cigar box completely filled with cassettes.

Now, one reason I chose the SWTP system is that software is plentiful and cheap. Unlike some other manufacturers' BASIC interpreters that sell for $100 or more, the SWTP BASIC costs less than $10. Even my box full of cassettes represented no more than perhaps $50 worth of tapes. Yet, I have used cassettes long enough in my stereo system that I worry about the eventual jammed tape, which is inevitable. It's a good idea to copy the tapes, use the copies and store the originals in a safe place.

Cassette Copying Methods—and Problems

There are various ways to copy computer cassettes. Since Kansas City Standard tapes use audio tones for digital data—eight cycles of 2400 Hz represents a digital 1, four cycles of 1200 Hz represents a digital 0—they can, theoretically, be copied like any other audio tape. With good-quality tape and two good recorders, I was able to play one tape and, at the same time, record the audio tones on another. This gave me acceptable, but not consistent, copies of tapes. Every now and then an error would creep in, much more often than I would have liked.

Another way to produce a copy is to read the tape into your computer and then write it back out. Since the tape is being produced by the computer, the quality should be every bit as good as the original. The computer can provide new clocking to compensate for speed errors in the original and it can provide a constant output level, even if the input has some level fluctuations.

There are actually several ways to use the computer in the process. The simplest is to load the cassette program into memory as if the program were about to be run, but then use the standard cassette write program to output it to cassette. Using the standard MIKBUG Load and Punch functions, I was able to produce excellent copies of program tapes in a short time with very little work.

But I soon discovered a flaw in the plan—MIKBUG formatted tapes use ASCII characters only, so an eight-bit byte is handled as two ASCII hexadecimal characters. This takes longer than if the tape were recorded in pure binary. Some of the longer SWTP tapes are supplied in pure binary format with a special loader on the cassette, but my copies were just plain MIKBUG format, and so took several times longer to load. I could have written a special dump program, but this seemed more work than it was worth.

Another way is to write a short program to read the original cassette tape and store each byte in memory exactly the way it appeared on the tape, including all the control commands, the loader program, as well as the pure binary code, and then write it back out exactly the way it came in. This would retain the exact format of the original and would load just as fast.

But this scheme has two serious flaws. First, it uses a lot of memory. In the worst case, where an entire program is in MIKBUG format, an 8K program may require more than 20K of memory to hold it in its uncompressed form. Second, when the tape contains a loader followed by the binary program itself, control characters are needed to disable the MIKBUG loader and jump to the loader just read. A short delay is needed before and after these control characters. The original tapes had them, but my copies did not. Back to the drawing board.

A third way of copying tapes and avoiding all of the foregoing problems is to write a short program that will input from the keyboard or reader,

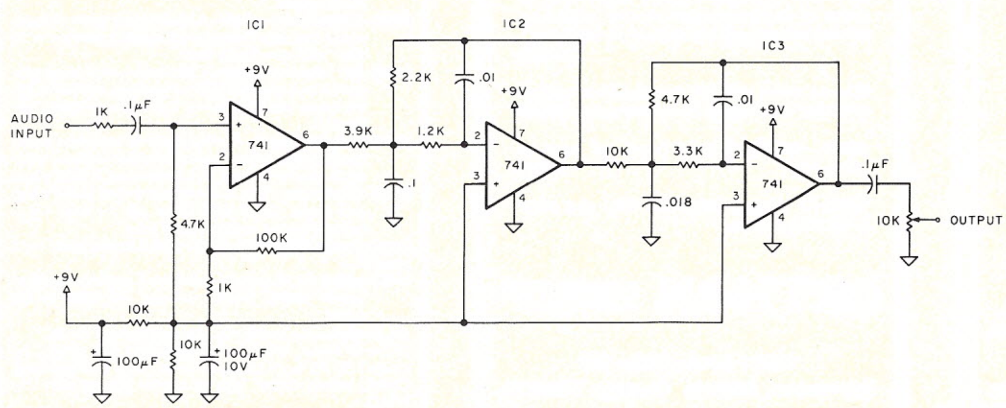


Fig. 1. Diagram of copying controller.

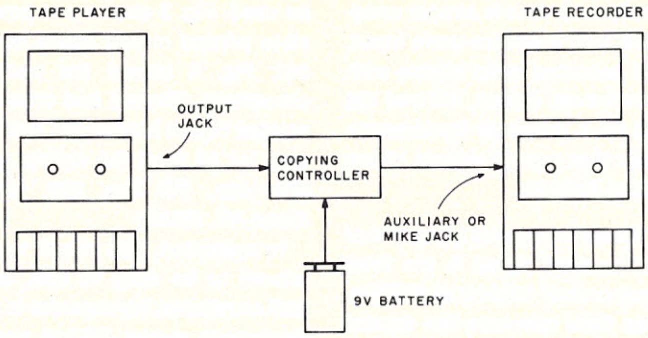
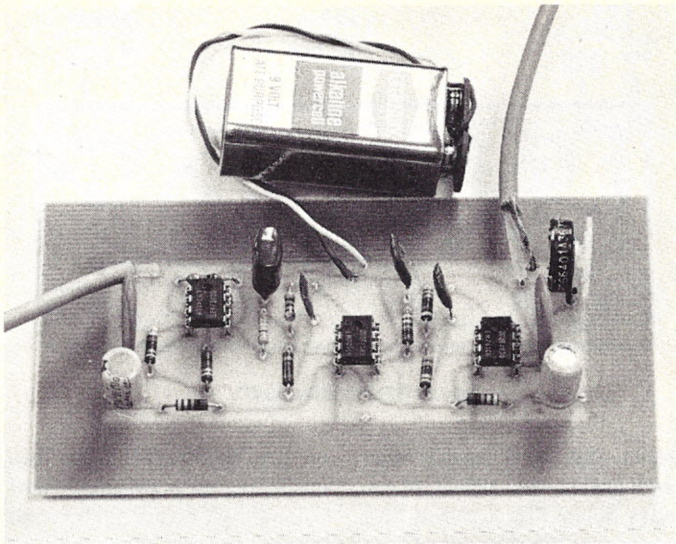


Fig. 2. Connecting the controller.



Tape copier board.

and immediately output to the printer or punch (a second cassette recorder in my case). In this way, I can read a tape and simultaneously write it out to another tape. All the timing of the original tape is retained, but any slight dropouts or level variations in the original tape are removed since the audio tones are again generated by the computer's cassette interface.

This is a workable and simple method, and only has two small disadvantages. First, any speed variations on the original tape or its tape player will be recorded onto the new tape, too. This is not too much of a problem, however, since Kansas City tapes are tolerant of wow, flutter, and other speed variations.

The major difficulty is that pure binary tapes can contain any kind of data, not just valid ASCII characters. Any one of the 256 possible 8-bit codes can occur for any byte. If the cassette interface or terminal is wired to respond to any of the special ASCII characters for functions such as starting or stopping cassette motors, switching input from cassette to keyboard and so on, then all hell will break loose as soon as a pure binary tape is copied.

After thinking about all these jolly possibilities, I decided there had to be an easier way to make a copy without having to analyze each and every case. The original idea of using two recorders to simply copy from one to another, just like any other tape, began to seem more attractive.

Analyzing the problem, I decided that there are two primary factors that can affect the quality of a cassette—speed or frequency variations and amplitude variations. Whenever the tape speeds up on playback, the frequency of all the tones increases proportionately. Tape-recorder wow and flutter are caused by speed variations; wow is caused by slow changes in speed, while flutter is caused by fast changes.

If you have a good ear for music, you can easily hear speed variations in a tape player that plays single tones or very slow music. Fortunately, the reference clock in the Kansas City tape format is recorded on the tape as part of the data, so that the computer can tolerate speed variations up to 30 percent or more from the normal. If the tape speeds up or slows down, the clock frequency also changes, and so the speed at which the computer's cassette interface decodes the data also speeds up or slows down to keep pace.

Amplitude variations are much more serious. Even though a tone may be recorded at a very precise level on a cassette tape (or even reel-to-reel tape), variations in the oxide coating of the tape cause the playback volume to vary over a large range. If the tape is poor or dirty, there may be times when the playback signal completely disappears; this is called a *dropout*. But even with high-quality tape, level variations still exist.

If you look at the output of the tape player, the output voltage can vary as much as 50 percent or so! If this happens, some of the bits on the tape may drop below the voltage sensitivity of the cassette interface, and be lost. The typical interface has enough gain to be able to compensate for this, though.

When we make a copy of such a tape using two recorders—even two good ones—the amplitude or level variations of the original tape are recorded on the copy, which also has its own level changes. Most of the time, these drops in level will occur at different times. But sometimes a small dropout in the copy may occur at the very spot where the signal is already weak because of a dropout on the original tape. In this case, the final signal level may drop below the point where the cassette interface can detect it, resulting in an error.

Solution

The solution is simple: control the volume during the copying stage so the dropouts do not get recorded onto the copy. To some extent, using a recorder with automatic volume control (AVC) may help, but most AVC circuits simply cannot respond fast enough to even out the rapid level variations caused by dropouts. Fig. 1 shows the circuitry of a simple device that solves the problem.

This is not just a level con-

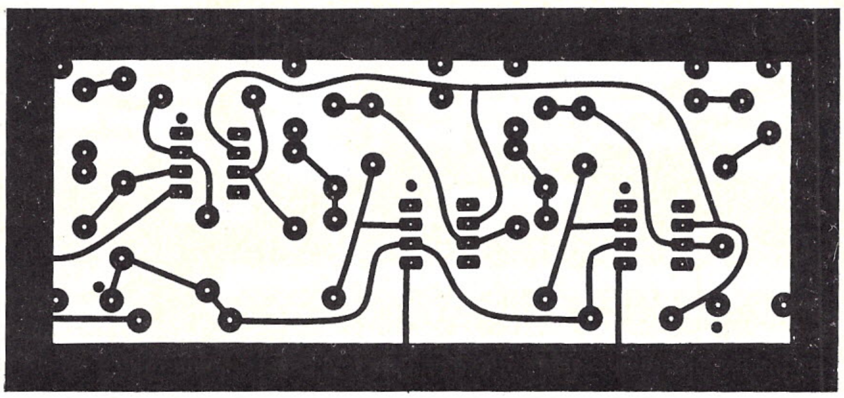


Fig. 3. Printed circuit board layout (copper side of board).

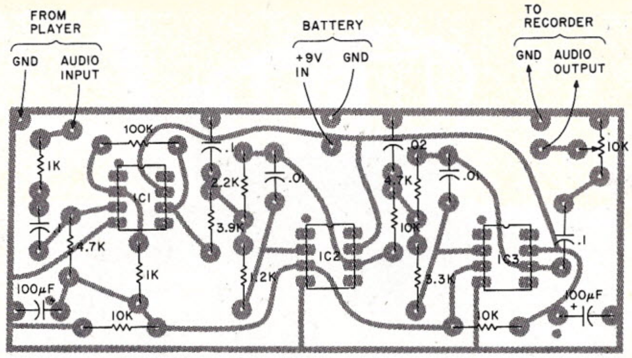


Fig. 4. Parts placement.

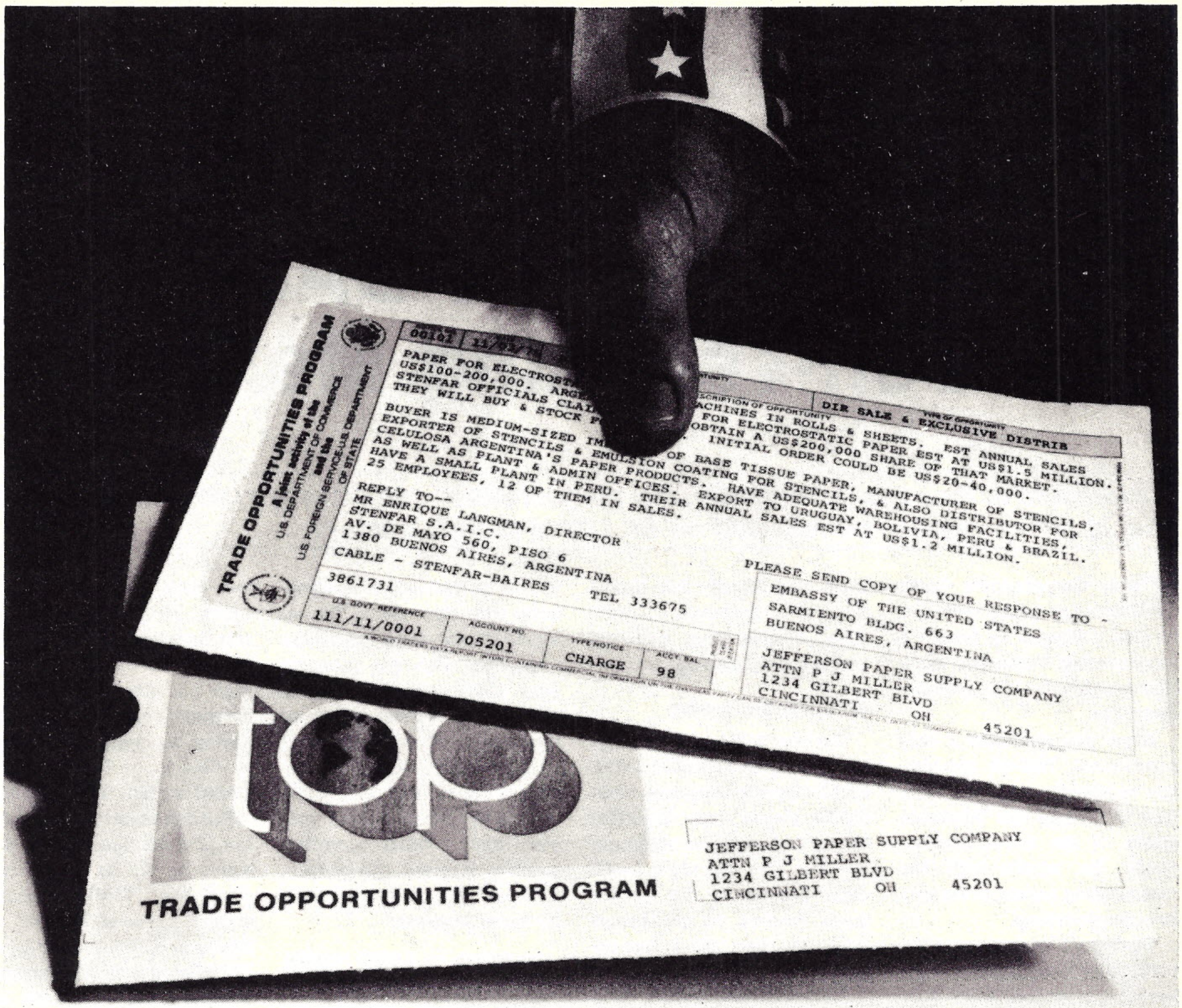
troller, but, in a way, it also regenerates the signal to remove noise. Fig. 2 shows how to connect it between two cassette recorders. The circuit consists of three inexpensive 741-type operational amplifiers. The signal is applied to IC1, which is a high-gain amplifier with a gain of 100. Enough signal is applied to the input so that this amplifier overloads and produces an output which is a square wave whose amplitude is constant, regardless of the exact input level. Because of the overloading of this amplifier, moderate amounts of noise simply do not get through this stage, and the output of IC1 is, therefore, a fairly clean square wave of either 2400 Hz or 1200 Hz.

The output of this stage could be applied directly to a recorder, but, from my experience, some recorders, especially those with automatic level controls, do not like square-wave inputs. They behave in strange ways and provide distorted output. One of my Panasonic recorders is almost unusable with a square-wave input, but becomes quite usable with sine waves. Hence, IC2 and IC3 act as a four-pole Butterworth filter that rejects signals above about 3000 Hz.

Since the lowest harmonic of the desired signal is the third harmonic of 1200 Hz at 3600 Hz, this filter converts the square wave into fairly respectable signals. 2400 Hz looks very much like a sine wave, and 1200 Hz is distorted but has smooth lines. Finally, a 10K potentiometer on the output allows adjustment of the output level to match either an auxiliary input or microphone input.

Construction is simple and the circuit layout or components are not critical. The only precaution is to avoid disk ceramic capacitors in the filter stages; use tubular, polystyrene or dipped capacitors instead. The circuit can be built on perforated board, or various other prototyping methods can be used. Fig. 3 shows one possible printed circuit board layout; Fig. 4 shows the parts placement. (Etched and drilled printed circuit boards are available for $5 from the author at Star-Kits, PO Box 209, Mt. Kisco NY 10549.)

In operation, this circuit has proven to be easy to use, as well as reliable. My recorder is a Panasonic model RS-314 portable, and the most common tape I use is Ampex type 350 in a C-60 size. The normal price locally is about 75¢ per cassette, though I have obtained it on sale for as little as 2 for 99¢. This combination has proven excellent—out of perhaps 15 or 20 tries at copying long programs like BASIC interpreters and assemblers, I have yet to make one bad copy. Unfortunately, Ampex has recently redesigned the mechanics of this cassette and the new version seems to be not quite as reliable. Several other tapes have proven to be just as good, though. ■



To get overseas trade leads like this, you could open offices in 127 countries.

Or use our computer.

The U.S. Commerce Department's computer-operated Trade Opportunities Program (TOP) can supply you with immediate, continuing, specific leads tailored to your sales objectives for any of 127 countries. So if you can't be all over the world at one time, we've got the answer. Send us the coupon now and see how you can find where the live leads are living.

Secretary of Commerce
U.S. Department of Commerce, BIC-9A
Washington, D.C. 20230

Please tell me more about the Trade Opportunities Program (TOP).

Name __________

Title __________

Company __________

Address __________

City _____ State _____ Zip _____

Ad Council

A Public Service of This Magazine & The Advertising Council

Sherman P. Wantz
424 NW Lakeview Drive
Sebring FL 33870

Something Extra With Radio Shack's BASIC: A Self-Teaching Manual

It's unheard of for us to get a review on a manual which comes with a piece of equipment. This is one of several we received on Dr. Lien's fantastic book!

Tandy Corporation's Radio Shack has a sure winner in its *User's Manual for Level I*, a programming instruction book that supports its TRS-80 microcomputer.

When you purchase a fully assembled TRS-80 system, you've already made the decision to forego the hardware phase of the microcomputer hobby. So, what's left? The never-ending aspect of the hobby—programming.

Teaching a newcomer to the hobby how to program in Level I BASIC is what the *User's Manual* is all about. For those who are new to hobby computing, let me explain that BASIC is an acronym for Beginner's All-purpose Symbolic Instruction Code, an interpreter that allows you, the operator, to communicate with your computer in English, rather than in binary code, the language that your computer is programmed to understand. I have bought and studied seven books that purport to teach BASIC to beginners. None of them comes close to equaling the short course that Radio Shack supplies with its TRS-80 system.

Dr. David A. Lien, leader of the writing team that produced the *User's Manual for Level I*, is an administrator in the California University system. His close association with the educational process is reflected in his writing. As you work your way through the 232-page book, you literally feel the instructor's presence . . . prodding, challenging, cajoling, praising and scolding.

A Word about Content

The *User's Manual* is a witty, fun-filled book written in a light vein for people who have never before placed their fingertips on the keyboard of a computer terminal. To make his material more palatable, the author uses humor in generous amounts. Here are a few examples:

"If you think GOTO is a powerful statement in everyday life, wait 'till you see what it does for a computer program."

"Play the (cassette) tape so you hear what digital data sounds like. You were expecting maybe Lawrence Welk?"

"How to goof-up FOR-NEXT loops." (by mis-nesting)

A few paragraph headings that establish a light tone for some otherwise heavy reading are:

"Debugging Programs. Quick—the RAID!"

"RND(X) with racing stripes."

"Caveat Emptor (Don't buy a used chariot from a stranger)."

The manual begins with a section on how to interconnect the video monitor, tape recorder, power supply and keyboard/computer—the four components that comprise the TRS-80 computer system.

Hookup instructions are easy to follow and assume that the owner has little or no familiarity with electronics. The instructions for using audio tape cassettes with the recorder even warn against trying to record on the leader portion of the tape. Pretty fundamental? . . . sure. But I'm impressed because I had to learn that simple lesson by losing a long program when I failed to run the audio tape leader into its take-up spool before I started to dump my program.

Inasmuch as BASIC resides in ROM (read only memory), the familiar READY message followed by an arrowhead (prompt) appears on the monitor's screen immediately after power is applied to the system. In simple, easy-to-comprehend language, the manual provides step-by-step instructions for clearing the monitor's screen, erasing the computer's memory, and confirming that the 3583 bytes (words) of memory are ready to accept data.

Before the TRS-80 owner is ready to begin learning to program (i.e., to make the computer do his bidding), he must become familiar with BASIC, the language that the TRS-80 was designed to understand.

Level I BASIC

It's almost impossible to evaluate the *User's Manual* without launching into a discussion of Radio Shack's Level I BASIC. After all, that is what the manual was written to teach. Hobby programmers who have been spending 4 to 15 minutes loading BASIC into their computers via paper or magnetic tape will appreciate having an interpreter resident in ROM. In the TRS-80 system, BASIC is ready when you are.

But can Level I BASIC do anything? It certainly can! It is so similar to the Dartmouth standard that TRS-80 owners should experience little difficulty in entering and running BASIC programs found in many computer hobby books and magazines. So, software programs should be in bountiful supply.

In addition to providing the usual BASIC commands, statements, functions and operators, Level I is loaded with extras. It contains four statements (SET, RESET, POINT and CLS) to control the graphics display. Yes, you read that right—the graphics display. SET lights (turns on) a specific spot on the video monitor's screen; RESET turns off the spot. When you specify the X-Y coordinates of a point on the screen, POINT reports whether or not that particular spot is being illuminated. CLS is used to clear the screen of all displayed data.

In addition, there are AT and PRINT MEM. AT is a print modifier (similar to TAB) that permits you to designate—by space number—where on the face of the screen you want to

print a message. PRINT MEM instructs the computer to report the unused number of bytes that remain in memory. It's nice—even necessary—to know whenever you are concerned that you might not have sufficient memory left to accommodate another variable in an array you are planning to use in your program.

Even the random-number generator found in Level I is an improvement over the type usually found in 4K BASIC. While RDN(0) still returns a random number between zero and one, RND(N) generates a random number between one and any number you may choose, up to 32,767. That's certainly superior to the PRINT INT((B - A + 1)*RND(0) + A) function that's found in most other BASIC interpreters.

Level I provides variables A through Z, A(X)—to store the elements of a one-dimensional array—and two string variables, A$ and B$, each of which is capable of representing 16 bytes. All right, so 26 variables and two string variables are limiting factors; I have seen 4K BASIC interpreters that offer no string variables, so I'm happy to have the use of A$ and B$.

In my discussions with Dave Gunzel, project manager for the *User's Manual,* I learned that Radio Shack is coming out with a Level II 12K BASIC in ROM for those who want to give TRS-80 greater capability than the Level I BASIC provides.

But back to Level I. Two logical operators are furnished: logical AND and logical OR. The AND operator is called by placing variables, math operators and relational operators inside parentheses separated by an asterisk; logical OR separates the parentheses with a plus sign. Here are examples of logical AND and logical OR statements:

AND: 40 IF (A=7)*(B=12) THEN 150, which means that if A equals 7 and B equals 12, the program should jump to line 150.

OR: 40 IF (A=7)+(B=12) THEN 280, which means that if A equals 7 or if B equals 12, the program sequence should jump to line 280.

Now, if those capabilities in 4K BASIC don't "blow your mind," how about the TRS-80's Level I shorthand dialect? Dialect is Radio Shack's term for the abbreviations provided for every command and statement. Instead of typing PRINT, LIST, RUN, FOR, NEXT, etc., you can type P. for PRINT, L. for LIST, R. for RUN, etc.

> "User programs included in the manual are meant to demonstrate the versatility of the TRS-80 . . ."

Besides saving time entering a program into memory via the keyboard, abbreviations save valuable memory space. And since the standard TRS-80 is supplied with only 3583 bytes of memory, conservation becomes important. Each time you type P. in lieu of PRINT, you save three keystrokes and three bytes of memory.

When you use abbreviated commands and statements and eliminate spaces, a program line can take on a strange appearance—for example, "40P.I:F.N=1TO100:R.A:N.N". In plain English, line 40 says: "PRINT I. For N equals 1 to 100. Read A. Next N."

Level I also provides multiple statements on one program line . . . you merely separate statements on the same line with colons. Nothing innovative, I admit, but it's definitely a bonus in 4K BASIC.

The error signal system used by Level I is unbelievably simple. When the computer doesn't understand a command or statement, it prints WHAT?, reprints the command or statement and places a question mark just *before* the detected error. When the computer receives an instruction that requires action beyond the system's capabilities, it prints HOW?, reproduces the faulty instruction and places a question mark just *after* the error. The third and final error signal is SORRY that appears on the video monitor's screen whenever the computer's memory limit has been reached.

Level I BASIC contains none of the advanced math functions (square root, trigonometry, exponentials and logarithms) you occasionally need. But, wait! Even though Level I does not have these functions intrinsically (built in), the *User's Manual* carries them as subroutines that you can store on tape and enter into computer memory as you need them. These math subroutines (including—would you believe?—arc-sine, arc-cosine and arc-tangent functions) allow the TRS-80 computer to be "educated beyond its intelligence."

User programs included in the manual are meant to demonstrate the versatility of the TRS-80, as well as to give the owner experience in entering and running programs that serve a useful and, in some cases, entertaining purpose. A few examples of user programs furnished with the manual are: a 12-hour clock with timing regulated by a FOR-NEXT loop; a school exam-paper grader; an amateur-radio cubical quad antenna design; a speed-reading test; a Dow Jones Industrial Group forecaster; a loan amortization plan; and a game of craps.

> ". . . as well as to give the owner experience in entering and running programs that serve a useful and, in some cases, entertaining purpose."

More about Content

In 26 chapters, the *User's Manual* takes the hobbyist from an introduction to programming in BASIC through the process of debugging (locating errors). Each chapter is replete with exercises aimed at requiring the novice to test his mastery of the programming material covered.

In most cases, the author poses questions and then answers them on the same page, thereby reducing reader frustration. More complex programming exercises to which answers are not immediately given are also presented. Solutions to these are presented in full in Part B of the three-part manual.

Most helpful to the fledgling programmer are the numerous programs that are presented and analyzed, line by line. This type of analysis explains the reasons behind selecting one command over another when a particular programming result is desired. The use of flowcharting as a programming tool is also covered.

But even as the author explains flowcharting, he gives this commonsense advice: "A flowchart is supposed to help you, not make more work than it's worth. When it stops helping and makes you feel like you're back in arts and crafts designing mosaics, then you've gone as far as the flowchart will take you (or more typically, you've passed its point of usefulness)."

Appendix C of the manual contains a combination function and ROM-test program that checks out all functions of the TRS-80—its ROM, its (RAM) memory and its graphics capability. A program to test system hardware is also included.

Whenever a new program fails to work, it's a common novice-programmer tendency to suspect that a computer hardware problem has developed. "If you're having trouble running a program and you think it may be the computer's fault, try this program," the

manual states. It adds: "If you don't get a BREAK message (or an infinite loop), you can relax about the TRS-80 and go back to troubleshooting your program." How's that for giving a new programmer the velvet-glove treatment?

Throughout the text, author David Lien uses the fiction-writer's technique—suspense—to keep his reader interested, hinting at the wonderful and potent forces he will be able to unleash as he learns to use this or that programming tool to be covered fully in the next chapter.

The author even instructs his readers to enter statements that will cause error signals to appear, to enter too few DATA statements, to use incorrect punctuation marks, etc.—to demonstrate the effects of input errors on the program. He believes that you learn by experience—both positive and negative. Making errors is certainly an unavoidable part of the human experience—as each and every programmer knows only too well.

Although there is not much in the book for a professional programmer (except, perhaps, to learn how a super instructor is able to communicate with his students), the *User's Manual* is worthwhile reading, even for the most experienced computer hobbyist.

Although I consider myself an intermediate-grade hobby programmer, at best, reading the manual, I learned many truths about BASIC that I had failed to grasp through my previous, unsupervised study.

For example, I didn't appreciate fully why 0.5 is added to a decimal number before it is rounded using an INT(X) function. I was also puzzled by the 0.2 I found added to a number before it was subjected to an ON-GOTO statement. Both, I learned from my manual, are used to avoid possibly disastrous rounding errors.

I didn't realize, either, that a trailing comma (a comma placed at the end of a line) can play havoc with a program. Nor did I appreciate fully that DATA statements could not share lines with other program statements (it never occurred to me to try that one). I learned the hard way that a false IF-THEN statement causes the computer to move to the next numbered program line instead of moving on to the statement that follows it on the same line. If I had only read Dr. Lien's book a year ago . . .

From the manual I even learned to pronounce GIGO (Gee-joe) which, if you haven't yet discovered, means garbage in-garbage out: If you feed the computer incorrect information, the computer will dutifully give you an incorrect output. Dr. Lien states the proposition this way: "Never before in the history of mankind has there been a machine capable of making so many mistakes so rapidly and so confidently."

What the Manual Isn't

The *User's Manual* is not a technical treatise on the design and operation of microcomputers, interface equipment or terminals. Nor is it a maintenance manual for the TRS-80. In fact, Radio Shack wants to discourage owners of the system from opening the computer or video monitor cases, preferring that they simply return all components to the nearest Radio Shack store for forwarding to a designated computer service center.

Because the TRS-80 is expected to derive its popularity from among those who have neither the time nor the inclination to undertake electronic equipment construction or repair projects, Radio Shack's policy of discouraging self-maintenance is understandable. However, the sheer volume of TRS-80 systems expected to be in use within the next few years makes it probable that Radio Shack will be deluged by requests for schematics, replacement parts and maintenance instructions. I was pleased to learn that the company is already working on a maintenance manual in anti-

cipation of that demand.

The only technical information in the *User's Manual* appears at the end of the book. It consists of one page of interface equipment specifications for the cassette recorder and the video monitor. Also included is a description of signals that appear on each of 40 pins available at the rear of the computer for mating to a connector. The 40-pin connector permits the TRS-80 owner to expand the capabilities of his system by attaching other peripheral components, such as printers, mini-floppy disks, modems, etc.

Manual Not Perfect

Nothing man-made is without flaws; the *User's Manual* is no exception. Physically, the book measures more than 22 inches when it is opened. That occupies a lot of table or desk space—more than I can spare adjacent to my computer. Furthermore, it's difficult to store such a long book on the bookshelf.

One redeeming feature of the manual is that its pages are bound by wide plastic rings that allow it to be folded in two and lie open in a reasonably flat position. I'd like to see the next printing of the manual change dimensions of the book (taller and narrower), eliminate some of the artwork and combine the numerous blank spaces on each page to make room for increasing the size of printing type used in Part C—"Prepared User's Programs."

I found more than 20 typographical mistakes, which I forwarded to Radio Shack's project officer for the TRS-80 manual. Considering that I received my TRS-80 some two months before the manual arrived (a 30-page preliminary user's manual came with the computer system), I can appreciate the pressure that Radio Shack managers applied to their writing team to get the final version of the manual into print and distributed. Normally, typographical errors can be passed off as sloppy proofreading and overlooked; but when typos occur in computer programs, they can cause some rather baffling problems, particularly for persons trying to learn programming.

Radio Shack is compiling a comprehensive list of errors found in its Level I manual and expects to distribute an errata sheet—possibly via its "Microcomputer Newsletter"—to all who have purchased the TRS-80 system. Dave Gunzel assured me that the list of corrections would be made available shortly—most probably before you read this announcement. If you own a TRS-80 and haven't yet received your copy, contact Dave at Radio Shack, Dept. 0025, Fort Worth TX 76102.

Summary

Anyone who has ever enrolled in a correspondence course can appreciate how difficult it is to master an unfamiliar subject while sitting at home, alone, reading from a book. To communicate computer programming concepts and techniques to those who have never before seen a computer takes a rare blend of teaching and writing skills. Throughout his book, author Lien demonstrates an unusual ability to simplify complicated ideas in a manner that instills confidence in the novice programmer.

In a personal note that appears on the first page of the *User's Manual*, Dr. Lien states: "This is not a conventional book. Every fair and unfair, conventional and unconventional, flamboyant and ridiculous technique I could think of was used." The result? A manual that is an excellent self-instruction course on programming worthy of the TRS-80 microcomputer system it was designed to supplement.

I urge each of you to find someone who has purchased a TRS-80 system and inspect the Level I manual. Judge for yourself the value of the contribution that Dr. Lien and his staff of writers have made to our growing community of computer hobbyists.■

Dr. Robert J. Cotter
The Johns Hopkins
School of Medicine
725 N. Wolfe St.
Baltimore MD 21205

The Amazing 1802: D/A and A/D Applications

1802 owners have been getting restless lately and complaining that there haven't been enough articles on their system in hobbyist publications. . . .

Several months ago I purchased a COSMAC Elf II microcomputer from Netronics. So far I've been absolutely delighted with its performance. Perhaps what drew me to it initially was its low cost. For slightly less than $100, this easily assembled kit, based on the RCA 1802 microprocessor chip, includes 256 bytes of random access memory (RAM), a keyboard, hexadecimal seven-segment output display, a TV monitor chip and a power supply on a single printed circuit board. In addition, 4K and 1K memory boards and an audio cassette interface board will be available shortly from the manufacturer and may be plugged directly into the original unit.

Although the initial kit is limited in memory capacity, the possibility of having enough computer capability to begin programming (without worrying about power supplies and spending a lot of money on I/O devices before I could learn how to use them) proved to be an attraction I couldn't pass up. The possibility of expansion made it a good investment.

Since that time I've been interested in how to interface my COSMAC with the analog world. I've found that I can do quite a lot with only 256 bytes of memory. In my own field of chemistry, I'm using my COSMAC for temperature programming and monitoring, generating scanning ramps for spectrometers and reading currents from electrodes in chemical solutions.

These functions all have in common the interfacing of the computer with a digital-to-analog converter (DAC) for producing programmable voltages, or using an analog-to-digital converter (ADC) to read voltages which may then be processed by the computer.

This article discusses some hardware designs for building 8-bit digital-to-analog and analog-to-digital circuits. Even though the 1802 is an 8-bit device, analog signals can be produced with 12-bit accuracy; the design of such a circuit is also included. Some programming techniques for producing voltage ramps and reading voltage levels are also discussed. Using these, the COSMAC can function as a scan generator, a digital voltmeter, or a function generator for producing sawtooth, triangle and square waveforms. First, a few words about the microcomputer itself.

The Microcomputer

The Elf II is an 8-bit microcomputer employing the CDP-1802 microprocessor. The microprocessor contains 16 general-purpose 16-bit scratchpad registers (labeled R0, R1, . . . R15), which can be used as memory pointers to address up to 65,536 (65K) bytes of memory, or as temporary storage for processing 8- or 16-bit words.

In the Elf II, the 256 bytes of RAM may be addressed using only the eight lower-order bits of the registers. In addition, three other 4-bit registers (labeled N, P and X) are used to select one of the 16 general-purpose registers. Also, an 8-bit D register is used for carrying out all logic and arithmetic functions and for moving bits in and out of registers. The Elf II has a 16 push-button keyboard for entering 8-bit binary words as 2-bit hexadecimal words.

As an example of the programming technique, a 1N instruction would be used to increment the general-purpose register pointed to by the number in register N. Therefore, a 12 instruction places a 2 in the N register, and the number in the 16-bit register R2 is then incremented by 1. Since the R2 register points to some location in memory, such an instruction might precede an instruction to fetch the next data value in memory. If the binary number in R2 were carried out through the DATA lines and converted to a voltage, this might be the first step in generating a voltage ramp. The 12 instruction is written operationally as R2 + 1.

Numbers may be moved from either the registers or from memory into the 8-bit D register preceding a logic or arithmetic operation. A 42 instruction, for example, moves the 8-bit number in the memory location pointed to by R2 into the D register. This is written as M(R2) → D. The contents of a register itself (because it contains 16 bits) must be moved in

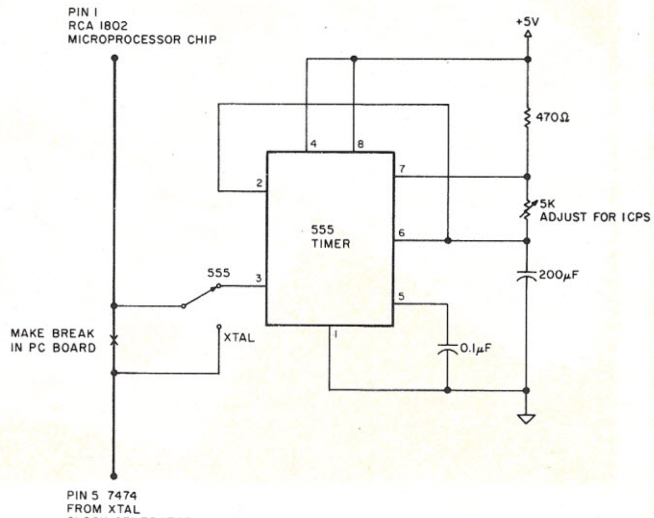


Fig. 1. 55 timing circuit.

two steps, so that an instruction 82 moves the lower-order bits of register R2 into D, or R2.0 → D, while a 92 moves the higher-order bits, R2.1 → D.

Two-byte instructions are entered one byte at a time. For example, the instruction 32 A0 can be entered from the keyboard by depressing 3, and 2, and then the INPUT switch, and then entering the second byte in the same manner. Two-byte instructions generally contain an instruction (32) and a direct memory location (A0). The example here performs a GO TO memory location A0, if the contents of D = 00.

The programming techniques are well described in the CDP1802 *User Manual,* and in several excellent articles by Joseph Weisbecker in the 1977 issues of *Popular Electronics.*

The Elf II comes with five sets of BUS lines that can be connected to any compatible input or output device. The BUS contains eight DATA and eight ADDRESS lines, clock and timing pulses, input flags, interrupt lines and three input/output enable lines.

For our D/A application only eleven connections are necessary. They include the eight DATA lines (D0-D7) and the enable lines (N0, N1 and N2), which allow the user to select one of three inputs or outputs. The number of devices can be expanded by decoding the N lines so that you may choose between devices represented by the eight numbers from 000 to 111. In addition, I have also made an output connection to the Q line for expansion of the number of outputs or inputs.

The computer comes with a 3.58 MHz crystal for producing the clock pulses. However, it can be slowed down to any speed, and I have found the timing circuit presented in Fig. 1 useful for checking both hardware and programming problems. The circuit uses a 555 timer and clocks at about 1 Hz; consequently, you can use a logic probe to follow the loading of binary numbers onto the digital-to-analog converter. Timing for the programs to follow, however, is based upon the 3.58 MHz clock frequency.

8-Bit Digital-to-Analog Conversion

The hardware for using the COSMAC Elf II as a programmable 8-bit digital-to-analog converter is shown in Fig. 2. The eight DATA lines (D0 through D7) are buffered using 4050 CMOS buffers. This not only protects the microcomputer CMOS circuitry, but also makes interfacing with TTL

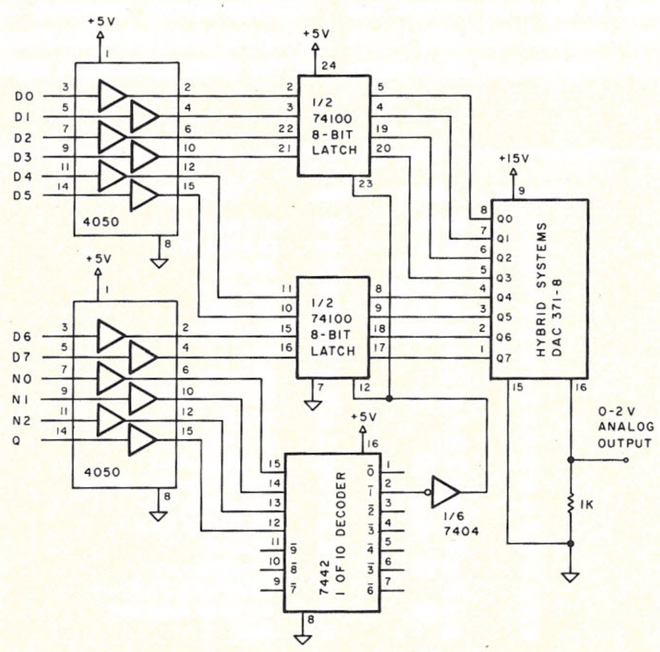


Fig. 2. 8-bit digital-to-analog interface.

Location	Bytes	Step	Comments
00	61 xx	01	xx → D/A converter
02	F8 xx A2	02	xx → R2.0
05	F8 yy B1	03	yy → R1.1
08	F8 A0 A3	04	A0 → R3.0
0B	21	05	R1-1
0C	91	06	R1.1 → D
0D	3A 0B	07	GO TO 0B if D≠0
0F	12	08	R2 + 1
10	82	09	R2.0 → D
11	53 E3	10	D → M(R3), 3 = x
13	61 23	11	M(R3) → D/A, R3 + 1, R3-1
15	FB zz	12	zz xOR D
17	32 1B	13	GO TO 1B if D≠0
19	30 05	14	GO TO 05
1B	7B 00	15	Q → ON, END

where: xx = starting voltage
yy = ramp speed
zz = ending voltage

Program A.

components possible.

The enable lines, N0, N1 and N2, are also buffered and then decoded using a 1-of-10 decoder, which acts as a 1-of-8 decoder as long as Q is low. The Q output may also be buffered and connected to the decoder, making it possible to select up to ten output devices with this decoder.

Whenever a 61 xx command is executed by the computer, the decoder sets its $\overline{1}$ output low and an inverter then places a high output on both halves of the 8-bit latch (pins 12 and 23 of the 74100). The latch then transfers the binary equivalent of the hexadecimal number xx to the DAC and holds that number until the next number is received. The particular DAC used in the circuit is really a binary-to-current converter, and the 1k resistor produces a 2-volt analog voltage whenever all the inputs are high.

At this stage we have built a programmable *voltage reference source* or a *constant current source* (without the 1k resistor). A single two-byte instruction, 61 FF, will produce 2.0 V or 2.0 mA; the instruction 61 7B will produce 1.0 V or 1.0 mA.

8-Bit Voltage Ramp

More often, however, it is important to program a series of voltages, which change over a period of time. When this is to be done, the 61 command can be made to output a number in a memory location that is constantly changing. A program for generating a voltage ramp is illustrated in Program A.

Step 01 loads the starting voltage into the digital-to-analog converter. Step 02 loads the same number into the eight lower-order bits of register 2, which will be incremented to give the 8-bit voltage ramp.

Step 03 loads the scan speed into the eight higher-order bits of register 1. This number will be decremented in a loop to determine the time between voltage steps. For example, if yy = 92, then the loop will take 1 second to execute, and the ramp will take 256 seconds to go from 0 to 2.0 volts. (Much faster speeds are possible if yy is loaded into the lower-order bits. In this case, the B1 in instruction 03 should be changed to A1, and step 06 should be changed to 81.)

Step 04 assigns a location, A0, outside of the program for register 3, which will hold the current number to be presented to the converter. Steps 05 to 07 form a loop in which the number yy is decremented and tested in the D register.

When the program exits from this loop, R2 is incremented in step 08. The new number (step 09) is sent to the D register and then to the memory location designated by R3 (step 10). The instruction E3 ties the X register to R3 so the 61 command will recognize this number as the value to send to the D/A converter in step 11. A 61 command also automatically increments register 3, so that a 23

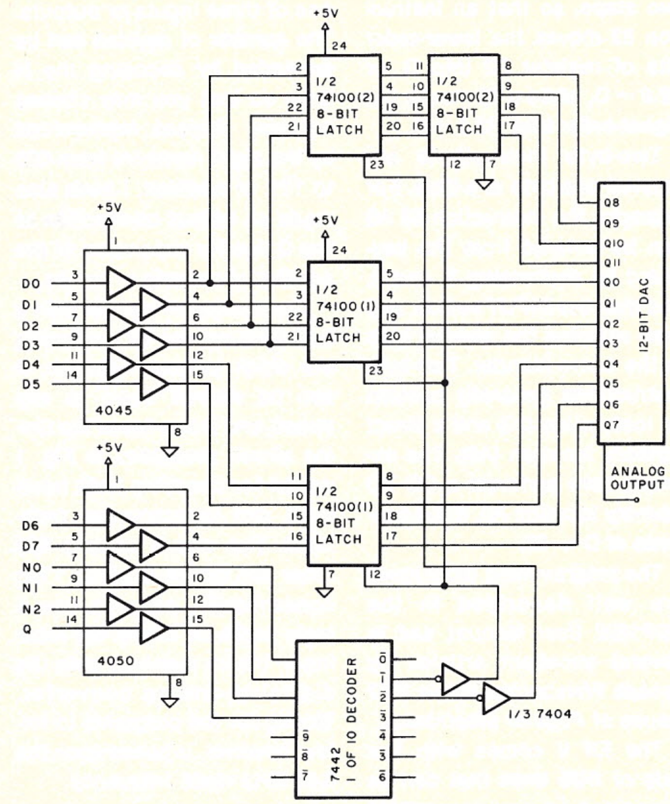


Fig. 3. 12-bit digital-to-analog interface.

instruction is necessary to keep the memory pointer at the same location.

Step 12 compares the current number with zz (using the Exclusive OR command), and step 13 will end execution if the final voltage has been reached. If not, step 14 will loop the program back for another voltage step. If the final voltage has been reached, then step 15 will light the on-board LED to signal the end of the program. If step 15 is changed to 30 02, the program becomes a *sawtooth generator,* with frequency, amplitude and offset determined by the values of xx, yy and zz.

12-Bit Digital-to-Analog Conversion

It is possible to make a 12-bit digital-to-analog converter by using the first twelve bits of one of the 16-bit scratchpad regis-

Location	Bytes	Step	Comments
00	62 00	01	0 → bits 8-11
02	61 00	02	00 → bits 0-7
04	F8 00 A1	03	00 → R1.0
07	F8 00 B1	04	00 → R1.1
0A	F8 A0 A2	05	A0 → R2.0
0D	F8 92 B3	06	92 → R3.1
10	23	07	R3-1
11	93	08	R3.1 → D
12	3A 10	09	GO TO 10 if D≠0
14	11	10	R1 + 1
15	91	11	R1.1 → D
16	52 E2	12	D → M(R2), x = 2
18	62	13	M(R2) → bits 8-11, R2 + 1
19	FB 10	14	R1.1 xOR D, → D
1B	32 23	15	GO TO 23 if D = 0
1D	81	16	R1.0 → D
1E	52 E2	17	D → M(R2),x = 2
20	61	18	M(R2) → bits 0-7, R2 + 1
21	30 0A	19	GO TO 0A (reset MX)
23	00	20	END

Program B.

ters. Bits 8-11 on the register can be sent to the DATA lines and latched by one output instruction to await the remaining bits. Bits 0-7 are then sent out to the DATA lines using a different output instruction, which then latches all twelve bits on the converter.

A circuit which accomplishes this is shown in Fig. 3 and can be made by adding another 74100 8-bit latch to the previous circuit and replacing the 8-bit converter with a 12-bit DAC. This circuit will also run the 8-bit program already described.

A 62 instruction is decoded and latches four bits (8-11) into one half of 74100(2). The second half of this latch is enabled at the same time the next eight bits (0-7) are received using the 61 output instruction. Thus, the instructions 62 13 and 61 57 load the 12-bit number 357 onto the digital-to-analog converter.

A *12-bit ramp* program is presented in Program B. It will scan the range from 000 to FFF in 4096 steps. The actual voltage range will depend upon the choice of converter used.

Programming for the 12-bit ramp is similar to that for the 8-bit. Steps 01 to 04 start the ramp at 000, but this time using both halves of register 1. Step 05 sets the memory location for the current voltage. Steps 06 to 09 determine the loop time for the voltage steps. The 92 in step 06 results in a ramp that takes well over an hour to execute and may be changed for faster or slower ramps. Step 10 increments the ramp register, and steps 11 to 13 are used to output the four higher-order bits. (The 62 instruction also moves the memory pointer to location A1, where the eight lower-order bits will be stored.)

Steps 14 and 15 are used to end the ramp when it has reached FFF. If the 10 in step 14 is replaced by 20, the ramp will execute twice; if 30, then it will execute three times, etc. This step can be used to produce a predetermined number of sawtooth cycles. Steps 16 to 18 output the lower eight bits, and step 19 returns the program for the next voltage step. Again, step 20 may be replaced with an instruction 30 04 to produce a sawtooth generator with 12-bit accuracy.

8-Bit Analog-to-Digital Conversion

Analog-to-digital conversion is generally a more complex process than digital-to-analog conversion. Most integrated circuit ADCs actually use a DAC that generates an analog voltage to be compared with the analog voltage to be measured. Some method is then used to sense when a correct comparison has been made, and the current digital number on the DAC is then recorded. Fig. 4 shows an addition that can be made to the 8-bit DAC circuit in Fig. 2 for analog-to-digital conversion.

The voltage output of the DAC is connected to the inverting input of a comparator. The analog signal to be measured is connected to the noninverting input. Whenever the voltage from the DAC exceeds the measured voltage, the output of the

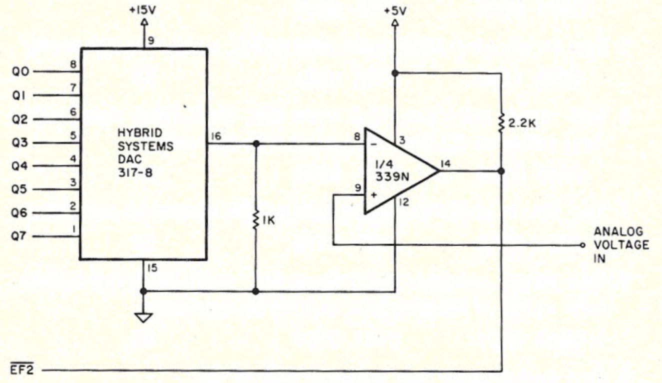


Fig. 4. Addition to 8-bit D/A circuit for analog-to-digital conversion.

Location	Bytes	Step	Comments
00	F8 A0 A2	01	A0 → R2.0
03	7B	02	Q → ON
04	F8 00 A1	03	00 → R1.0
07	52 E2	04	D → M(R2), x = 2
09	61 22	05	M(R2) → D/A, R2 + 1, R2-1
0B	35 11	06	GO TO 11 if EF2 = 1
0D	11 81	07	R1 + 1, R1.0 → D
0F	30 07	08	GO TO 07
11	64 22	09	M(R2) → HEX DISPLAY, R2 + 1, R2-1
13	F8 0F B3	10	0F → R3.1
16	7A	11	Q → OFF
17	23 93	12	R3-1, R3.1 → D
19	3A 17	13	GO TO 17 if D≠0
1B	30 03	14	GO TO 03
1D	00	15	END

Program C.

Location	Bytes	Step	Comments
00	F8 A0 A1	01	A0 → R1.0
03	F8 80 A2	02	80 → D; D → R2.0
06	51	03	D → M(R1.0)
07	E1	04	M(R1.0) = MX
08	61 21	05	MX → converter
0A	64 21	06	MX → hex display
0C	3D 11	07	GOTO 11 if EF2 = 0
0E	82	08	R2.0 → D
0F	F5	09	M(R1.0)-D → D
10	51	10	D → M(R1.0)
11	82	11	R2.0 → D
12	F6	12	shift D right
13	32 19	13	GO TO 19 if D = 0
15	A2	14	D → R2.0
16	F4	15	MX + D → D
17	30 06	16	GO TO 06
19	64 21	17	MX → hex display
1B	00	18	END

Program D.

comparator swings low. The output is connected to one of the input flag lines of the computer, in this case the $\overline{EF2}$ line. The instruction 35 tests EF2 to see if it is high.

Using this circuit, two basic methods can be used to determine the analog voltage. The first method uses a *voltage ramp,* which starts at 00 and stops whenever the digitally generated voltage has just exceeded the voltage to be measured. It is the slowest of the two methods because as many as 256 comparisons may have to be made. Also, the acquisition time will be dependent upon the voltage to be measured, since a lower voltage will need fewer comparison steps. However, the maximum acquisition time for the program (Program C) is around 10 ms and should not present a problem for slowly changing voltages.

The other method uses a *successive approximations* technique, in which each of the eight bits is tested once (starting from the highest order bit) and retained if it does not exceed the voltage to be measured. The acquisition time is always the same.

The combined circuit from Figs. 2 and 4 can be used to measure any voltage between 0 and 2.0 volts. Program C uses the ramp technique and continuously displays changing voltages on the Elf's LED outputs. It simulates the operation of a digital voltmeter (DVM).

Step 01 sets the memory location that keeps track of the digital number on the DAC. Step 02 lights the on-board LED to indicate that the voltage is being acquired. If this feature is used, then it is necessary to ground pin 12 of the 1-of-10 decoder so it will not decode the status of the Q line. Step 03 starts the ramp at 00. Steps 04 and 05 send this number (and later numbers) to the DAC.

Step 06 tests the EF2 line and sends the digital number to the output display if a correct comparison has been made. If not, steps 07 and 08 increment the digital number and return the loop. Steps 10 to 13 form a timing loop to introduce some delay between measurements. This is necessary to keep a number on the display long enough to see it. The value 0F in step 10 results in a measurement about every tenth of a second. Step 11 turns the LED off whenever acquisition is not taking place. Step 14 loops the entire program so it will give continuous voltage readings.

If you have built your circuit correctly, you can then connect a 10k potentiometer between your 5 V power supply and ground. The center post of the potentiometer is then connected to the analog input of the ADC circuit. As you rotate the potentiometer shaft, the hexadecimal readings on the computer display will change. If you connect a voltmeter between the center of the potentiometer and ground, you can then get an idea of the correspondence between the hexadecimal readings and voltage. For example, 0.5 V will give a reading of 40; 1.0 V a reading of 80; 1.5 V a reading of C0, etc.

If the circuit is operating correctly, you may want to try a program for analog-to-digital conversion using the successive approximation method. Enter the program shown in Program D.

Step 01 sets the memory pointer R1.0 to location A0. Step 02 places the hexadecimal number 80 in register 2. This corresponds to the binary number 1000 0000 and is used to test the most significant bit of the ADC. Register 2 will generate a binary number for each bit by shifting to the right each time the program goes through one cycle. In step 03, the same number is loaded into memory location A0, which will accumulate the sum of all bits that give a voltage less than the measured voltage.

Step 05 places 1000 0000 into the converter, and in step 06 the number 80 is displayed on the hex display. This step is not necessary, but is used to observe the successive approximation in progress when the computer clock is slowed down. Step 07 tests the comparator in the ADC circuit.

R2.0	MX	D/A(displayed)
1. Load 80 into R2.0 and MX; output to converter		
1000 0000	1000 0000	80
2. Too large; subtract R2.0 from MX; shift R2.0 right; add new value of R2.0 to MX; output to converter		
0100 0000	0100 0000	40
3. Too small; shift R2.0 right; add to MX; output to converter		
0010 0000	0110 0000	60
4. Too large; subtract R2.0 from MX; shift R2.0 right; add new value of R2.0 to MX; output to converter		
0001 0000	0101 0000	50
5. Too small; shift R2.0 right; add to MX; output to converter		
0000 1000	0101 1000	58
6. Too large; subtract R2.0 from MX; shift R2.0 right; add new value of R2.0 to MX; output to converter		
0000 0100	0101 0100	54
7. Too small; shift R2.0 right; add to MX; output to converter		
0000 0010	0101 0110	56
8. Too large; subtract R2.0 from MX; shift R2.0 right; add new value of R2.0 to MX; output to converter		
0000 0001	0101 0101	55
9. Exit and display last value of MX = 55.		

Table 1. Successive approximation determination of 0.66 V.

If the number 1000 0000 is too large, steps 08-10 will subtract the value in R1.0 from this number. Steps 11-15 will shift R1.0 to the right and place the number 0100 0000 in MX in step 03. If the number is too small, then the program will jump to step 11, shift R1.0 right and place the number 1100 0000 in MX. When all eight cycles have been completed, the last number in MX, which corresponds to the analog voltage, will be placed in the hex display of the computer. Table 1 shows the sequence of events in the measurement of 0.66 V at the analog input.

If you have built the 555 timing circuit, you can watch the sequence of successive approximations on the hex display. Also, you can connect a voltmeter across the output of the DAC and watch the corresponding analog approximations. Three examples, 0.66 V, 0.95 V and 1.40 V, are measured in Fig. 5. The voltage steps correspond to the output of the DAC, while the numbers printed above each step correspond to what you will see on the computer hex display.

When the computer is run with the crystal clock, the acquisition takes place in less than 0.6 ms and is reasonably constant for any voltage value measured. Of course, you may also replace the last step in the program with a delay loop and a return to step 02, and you will again have a digital voltmeter.

Summary

Once you have built your D/A and A/D converters, you will find an infinite number of uses for your computer. (By the way, I recommend building all of your output circuitry on one of those solderless IC breadboards, since you will want to make changes in the hardware almost immediately to fit your own applications. I used a Prototoboard 203A, which comes with ±15 V and +5 V power supplies, so I can add as many op amps or logic devices as I wish.)

Your digital-to-analog converter can be used to program changes in a temperature device or to control the speed of your model railroad. You can write a program that produces a square wave at any frequency and duty cycle, if you output the high and low voltages, separated by timing loops. The ADC circuit makes your computer a digital thermometer when connected to a suitable temperature-sensing device.

Perhaps the most interesting applications are those in which the analog input data is processed by the computer, and not merely read out through the hex displays. You can read data from two voltage sources (using the N0, N1 and N2 lines), store the values in memory, subtract one from the other and display differential readings. Or you can subtract successive readings from one device and plot a derivative trace on a recorder or scope using the DAC. You can accomplish integration by adding successive readings in the D register and multiplying by the sampling interval.

Rapid changes in voltage can be stored in a large number of memory locations and read out slowly at some later time. For this task, 256 bytes of memory is not enough; you will probably then want to write for *Kilobaud* so you can afford that 4K memory board and cassette interface board. ■

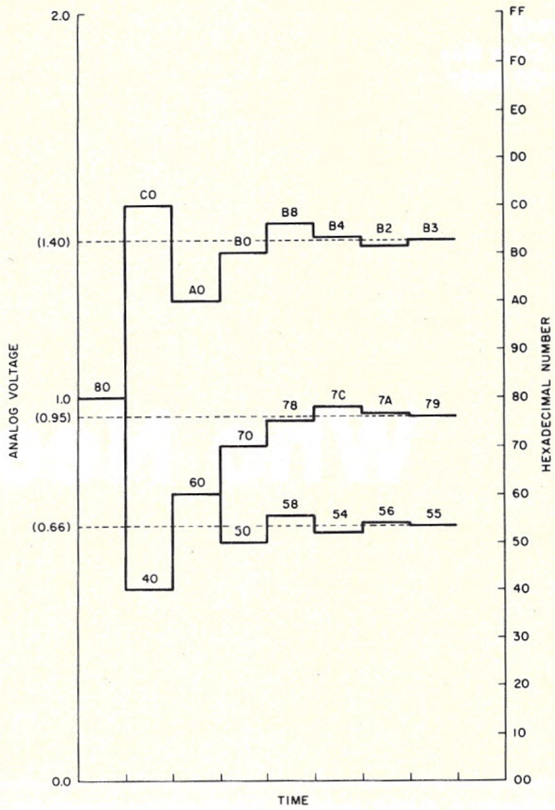


Fig. 5. Successive approximation examples.

F. J. Greeb
1915 S. Cape Way
Denver CO 80227

Who Needs a UART?

It's always nice when someone comes up with a less expensive method of doing something. UARTs being as costly as they are, you might be interested in this alternative.

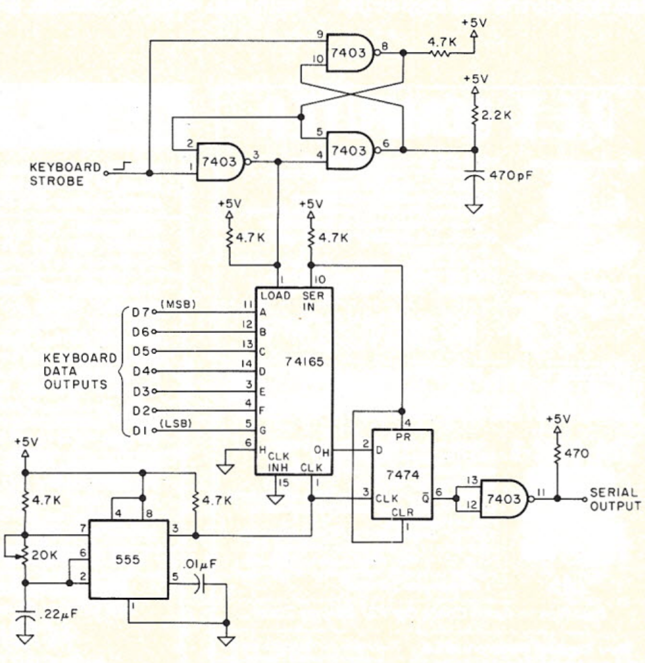
Why use a UART or a similar special-purpose chip to interface the parallel outputs from your keyboard to a serial input port? The circuit shown here accomplishes the same function using only four standard chips, including the clock circuit. These chips are easy to find, and their total cost is less than that of a special-purpose unit. The circuit can be wired on a small piece of perfboard and installed in the keyboard case. Only 5 volt power is required, which can be taken from the same source used to power the keyboard.

The keyboard output data is fed to seven of the parallel inputs of the 74165 shift register; the eighth input is tied low. Three of the 7403 gates are wired to form a one-shot, which converts the keyboard strobe to a narrow pulse to load the shift register. The fourth gate in the package is used as an output buffer.

On the first clock pulse following the load action, the data bit that is tied low will be shifted out through the 7474 flip-flop, providing the required start bit. The seven data bits are then shifted out in sequence. The serial input to the shift register is tied high so that after the data bits have been transmitted, the output is always high, thus providing the required stop bits. The timing components shown for the 555 timer clock circuit allow adjustment over a range of approximately 150 to 1400 Hz. This allows operation at standard baud rates of 150, 300, 600 or 1200. The shift clock can also be derived from the computer by installing a 7493 counter chip on the interface board to divide the normal UART clock by 16. This allows the data rate to be changed at the computer without requiring any adjustments of the keyboard interface circuit.

For a negative strobe keyboard, the fourth gate in the 7403 package can be used as an inverter in front of the one-shot circuit. This removes the output buffer, which is not as desirable, but should not present any problems if the line lengths are kept short. Eight data bits can be handled by using the second flip-flop in the 7474 package. Wire the eight data bits to the shift-register inputs; connect the one-shot strobe pulse to the clear input of the first flip-flop; and add the second flip-flop as a new stage in the shift sequence.

Note that it is theoretically possible to load new data into the shift register before the last character has been shifted out by pressing two keys in rapid succession. In practice, this will rarely, if ever, happen. I have been using the interface for several months, operating at 1200 baud, and have not experienced this problem. The output is connected to a serial input port set for eight data bits and one stop bit. As a result, the eighth data bit is always received as a high level. ■



Circuit diagram. Keyboard parallel to serial conversion.

Can't Find It?

SWTP BASIC owners and users who have had this problem with their manuals can copy this handy index to save search time.

Charles E. Thomas
7022 Blackhawk
Pittsburgh PA 15218

Even though the SWTP 8K BASIC (version 2.0) manual consists of only 27 pages, I am constantly leafing through it to find a particular function or definition. However, the publisher left out one of the most important parts of the manual. The index page is missing!

I produced this index page for my local computer club's 6800 user's group. Make a copy of the index and stop wearing out the pages of your manual by leafing through looking for information. ■

Index.

COMPUTER SHOPPING ?

FIND YOURS! — In the comprehensive index of computer firms, The Computer Data Directory. Contains profiles of over 600 manufacturers, stores, distributors, publishers, and electronic dealers. All addresses are up-to-date and at your finger tips! Products are cross-referenced to their manufacturers. Similar to a telephone directory. Definitely a useful reference source. Order yours today. All orders shipped on day received.

ORDER NOW!
Only $4.98
Postage Included

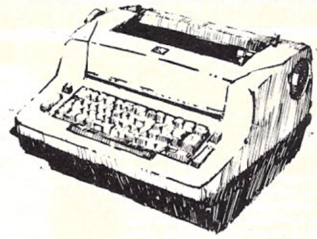
- ☐ SYSTEMS
- ☐ SOFTWARE
- ☐ PERIPHERALS
- ☐ ACCESSORIES
- ☐ EASY TO USE!
- ☐ COMPUTER STORES
- ☐ COMPUTER SERVICE
- ☐ ELECTRONIC PARTS/SERVICE
- ☐ BRAND NAME MANUFACTURERS
- ☐ WHERE TO WRITE FOR BROCHURES!

C77

THE Computer Data Directory

BOX 598 · DEPT K   CLEVELAND, OHIO   44107

SUPER SALE



$199.95
AS IS

$249.95
WORKING & CLEANED

IBM® Selectric-Based I/O Writers
Excellent Hobby Printers

Series 72/731
Heavy Duty
8½" Platten
All Solenoids
Original Documentation
While Supply Lasts

These terminals are from a large airline reservation system. They are heavy duty and were under continuous maintenance. The units have been in storage. We make every effort to ensure that all essential parts are included. Most work when plugged in. No warranties are given or implied.

Conversion Kits.
1. Conversion instructions, P C board for printer only using software method $59.95
2. I/O kit, instructions and all components for parallel or RS232 interface $249.95
3. Completely converted unit and assembled interface $999.95

Check, Money Order, Cash. Personal checks require 3 weeks to clear. No COD's. Units shipped UPS or PP collect. Prices Net FOB Tulsa

SUPER SURPLUS SALES S55
P.O. BOX 45944
TULSA, OK 74145
1-918-622-1058

DELTA electronics D27

590 Commonwealth Ave. Boston, MA 02215

(617)267-5900

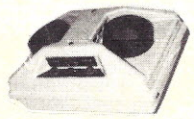
PHONE OR MAIL ORDER (617) 267-5900

WE ACCEPT: VISA, BAC, MC, CHECKS, OR MONEY ORDERS...INCLUDE APPROX. 10% EXTRA FOR SHIPPING/HANDLING CHARGES...EXCESS REFUNDED...

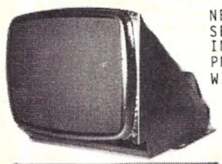
VISIT US AT OUR NEW LOCATION IN BOSTON!! OVER 5000 ITEMS IN STOCK....

PROFESSIONAL-PERSONAL (NEW) COMPOSITE VIDEO MONITORS..12" CRT, 14 MHZ, 110 OR 220 VAC, COMPACT, SIMULATED WALNUT-GRAINED CABINET ALL CONTROLS ON SIDE. $174.50

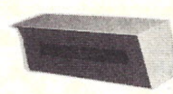
NEW EXECUTIVE KEYBOARDS ! 72 KEYS..SOME LIGHTED.. NON-ASCII, EASY TO PERSONALIZE WITH AN ASCII ENCODER CHIP.SIMULATED WALNUT GRAINED, (MATCHES MONITOR) $44.95



RECONDITIONED, COMMERCIAL ACOUSTIC COUPLERS (RS232) FULL AND HALF DUPLEX..... 110-300 BAUD, TEST MODE.. CARRIER DETECT....$169.50



CRT MONITOR
NEW COMMERCIAL QUALITY SETCHELL CARLSON TTL.. INPUT, P39 OR P4...... PHOSPHOR. 18MHZ BANDWIDTH..........$148.50

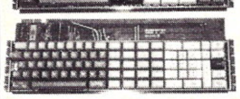


BURROUGHS (NEW) SELF-SCAN ALPHA NUMERIC DISPLAYS... 16 CHARACTERS WITH 5X7 DOT MATRIX..HIGH CONTRAST..112 COLUMN..WITH APPLICATION DATA.... $79.50

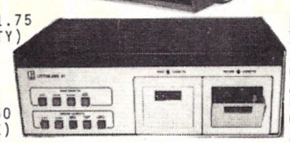
NEW ASCII KEYBOARDS..... MICRO-SWITCH (WITH DATA)

53 KEYS...........$49.50 (ALL TTY CODES)

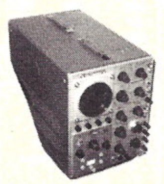
78KEYS............$61.75 (PROFESSIONAL QUALITY)

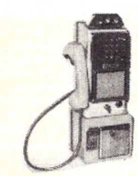


82 KEYS........$64.50 (EASY TO PERSONALIZE)



NEW LITTON ABS91 COMMERCIAL DUAL DRIVE DIGITAL CASSETTE SYSTEM....INCLUDES CABINET, CONTROLS, POWER SUPPLY, AND ALL LOGIC...50 CPS I/O TRANSFER...8 BIT PAR., DATA RECORD 11 BIT SER. STORAGE 50K X 8/DRIVE..EXTENSIVE MANUALS ....$670.00!!





POWER SUPPLY SPECIALS!!!

VOLTS	AMPS	NOTE:	STOCK #	PRICE
+5	4.0	RTL	B301	$29
+5	10.5	RTL	B302	$55
+5	12.6	RTL	B312	$65
+5	13.0	RTE	B314	$62
+5	16.0	RTL	B315	$82
+5	20.0	RTL	B316	$95
+5	31.0	RTL	B317	$135
+5	24.0	RTL	B318	$98
+5	32.5	RTE	B319	$138
+5	74.0	RTP	B320	$198
+5	74.0	RTL	B321	$198
+12	2.5	RTL	B323	$21
+12	5.0	RN	B324	$38
+12	7.3	RTL	B325	$44
+12	10.0	RTL	B327	$75
+12	10.5	RTE	B328	$65
+12	16.0	RTL	B329	$90

VOLTS	AMPS	NOTES:	STOCK #	PRICE
+12	26.5	RTL	B330	$144
+15	9.5	RTL	B331	$90
+6	5.0	RN	5451	$29
+13.5KV	.31MA	RN	5356	$14
+10 TO +20	1.5	RNPV	5311	$25
+90	1.0	XN	5154	$14
+24	20	XN	B332	$48
+4.25	150	RNEH	B336	$137
+12,-12	5.0			
-36	1.0			
+4.6 TO6.8	0-70	RTV	B338	$195
+12 TO +15	.4	RTL	B339	$23

R = REGULATED
X = UNREGULATED
N = NEW-GUARANTEED
T = TESTED & GUARANTEED

V = VARIABLE
H = 220 VAC
L = LAMBDA (MFR)
E = NORTH ELECTRIC (MFR)
P = POWER MATE (MFR)

TEKTRONIX 310A SCOPE: RECONDITIONED/GUARANTEED $385
TEKTRONIX 545A SCOPE: RECONDITIONED/GUARANTEED $575
TEKTRONIX 535A SCOPE: RECONDITIONED/GUARANTEED $395
DUAL TRACE PLUG-IN "CA"........................ $160
FAST RISE "L" PLUG-IN $94

NEW PRECISION JOY STICKS WITH 4 100K POTS 2 FOR$9.50

NEW EKG STRIP CHART RECORDER w/EVENT MARKER.... $95

FACTORY RECONDITIONED TOUCH TONE PAY PHONES WITH KEYS.............................. $85

VISIT OUR NEW STORE AND MAIL ORDER WAREHOUSE NEAR KENMORE SQ., BOSTON.

WHEN PLACING ORDERS FOR POWER SUPPLIES, MAKE ALTERNATIVE CHOICES IF POSSIBLE...

Professionally developed software for the PET. Extensive use of dynamic graphics such as running horses, spinning slots, and playing cards illustrated on screen. Also financial analysis, amortizations, etc. Order now and be placed on mailing list for future news and releases.

Let us know your preferences.

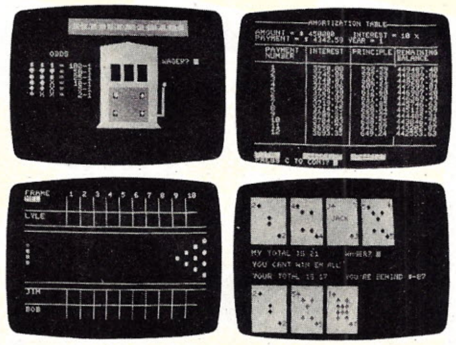
CASSETTE 2: Slots, Blackjack, Poker, Craps
CASSETTE 3: Golf, Bowling, Hamurabi, Robots, Checkers
CASSETTE 7: Financial Analysis and Amortization Schedules

Price $14.95 each cassette plus $.75 shipping & handling. 10 other cassettes available. PDP-11 cards to interface with line printers, card readers, mag tape and CRT — now available through Eclectic. Extensive PDP-11 software also available.

Write for more details. Or, order by phone: 214/358-1307. VISA and Master Charge accepted.

ECLECTIC CORP.
2830 Walnut Hill Lane
Dallas, Texas 75229

E32



Eclectic Software
2830 Walnut Hill Lane
Dallas, Texas 75229
214/358-1307

Name ______________________
Address ______________________
City ______________________
State ______________ ZIP ______
Bankamericard/VISA ______________
Master Charge ______________
Bank Card No. ______________
Expiration date ______________
Signature ______________________

LOW COST VIDEO TERMINAL

Shown with Zenith 9" TV

QUIETLY REPLACE THAT MECHANICAL TELEPRINTER

Connects to any transformer powered B&W TV with a simple video input modification. Or, we can provide a Zenith 9" set with the necessary modification that can also be used for TV reception.

* Finest quality long life keyboard
* 16 line by 64 character display
* Upper case ASCII character set
* Auto scroll, auto line feed, and auto return
* Selectable Baud rate, 110 or 300 full duplex
* Interface, both RS-232C and 20mA current loop
* Quartz crystal timing
* Sturdy metal case
* Cursor

Fully assembled Video Terminal	$350
Modified Zenith 9" TV	$100
(Available only with terminal)	

Call or write for additional information

MC and VISA ACCEPTED

MICROTYPE INC.
404 E. Park, Plano, IL. 60545 ● (312) 552-7074

M50

TRS-80
MACHINE LANGUAGE SOFTWARE

MICROCHESS: PLAY CHESS AGAINST YOUR TRS-80 - $19.95

MICROCHESS plays chess at 3 levels of difficulty. Although it may not beat the chess masters, it does play an aggressive game and is not easily beaten. In addition to being great fun to play, it provides a useful and tireless opponent for practicing checkmates, learning openings, and for sharpening general playing skills. You may set up the board any way you wish. You can switch sides with the computer at any time. You can even make the computer play against itself! Written by Peter Jennings, MICROCHESS has held its own against much larger chess-playing computers. It is a truly unique program for the TRS-80!

RSM-1S: A MACHINE LANGUAGE MONITOR FOR THE TRS-80 - $23.95

RSM-1S provides you with 22 commands with which you can interact directly with the Z-80 processor in your TRS-80. You will have direct access to all memory locations. You may examine your Basic ROM's, test your RAM, enter and execute machine language programs, read and write machine language tapes, and much more!

A symbolic dump command disassembles object code in memory and displays it as Zilog standard Z-80 mnemonics! Memory may also be displayed in HEX or either of two ASCII formats. Memory can be EDITED, MOVED, EXCHANGED, VERIFIED, FILLED, ZEROED, TESTED, or SEARCHED for one or two-byte codes. Memory display commands may be stepped one line at a time with the space bar, or may be terminated by use of BREAK. RSM-1S is a unique and superior monitor!

RSL-1: GRAPHIC PATTERN DRAWING AND THE GAME OF LIFE - $14.95

With RSL-1 you can draw graphic patterns on your display directly from the keyboard, or you can load patterns from cassette. The keyboard has a unique repeating function that will write a continuous line in eight vertical, horizontal, or diagonal directions! Patterns may be saved on tape, and four demonstration patterns are furnished on your RSL-1 cassette. After the pattern is entered, RSL-1 plays LIFE, a game of birth, growth and death of a colony of cells. RSL-1 will compute and display each generation for about 2 seconds, regardless of the pattern on the screen!

OTHER TRS-80 PRODUCTS

RSM-1: $17.95. Identical to RSM-1S, but without symbolic display command.
ESP-1: $29.95. A resident assembler and editor using Intel 8080 mnemonics.
LST-1: $7.95. A disassembled listing of LEVEL-1 BASIC with some comments.

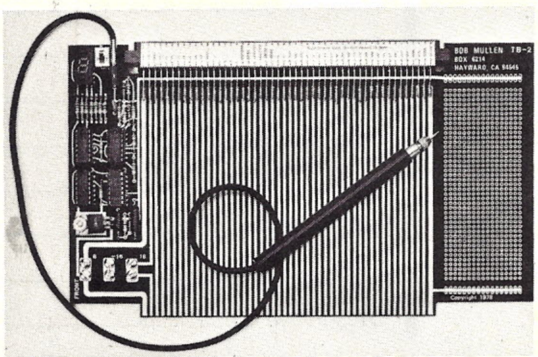
All software runs in both LEVEL-I and LEVEL-II 4K or 16K systems. Immediate delivery! Order yours today! (California residents add 6% sales tax).

* SMALL SYSTEM SOFTWARE * POST OFFICE BOX 483 * NEWBURY PARK, CALIF. 91320 *

S51

YES, it's improved. YES, it's the same price!

CAVE grafix™

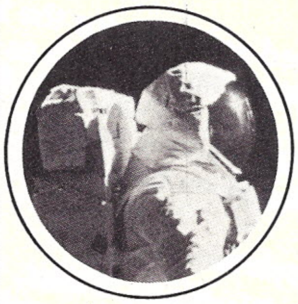


At $35, the Mullen Extender Board Kit with built-in logic probe has always been a great value . . . but when it came time to crank up a new production run of boards, we couldn't resist adding a few small changes here and there to continue staying ahead of the pack.

For example, our logic probe now reads out into a 7 segment readout; we've also included a pulse stretcher and a light whose brightness is proportional to the duty cycle of a pulse stream (activates when probe reads H or high). There's a general-purpose breadboard area where you can develop circuits unique to your needs, with holes punched on a .1″ grid. The edge connector is not only easier to mount and virtually self-aligning, but also allows use of scope hooks . . . we could go on, but that's enough for now. **Check out the new Mullen Extender Board at your local computer store today, or order by mail** (Cal res add tax) **from:** M32

MULLEN COMPUTER BOARDS

BOX 6214, HAYWARD, CA 94545



SPACE AGE COMPUTER STORE

NEW YORK CITY
and
LONG ISLAND

IF YOU WANT PROFESSIONAL SERVICE IN A CASUAL ATMOSPHERE . . . AND A LARGE VARIETY OF EQUIPMENT

- **BUSINESS APPLICATIONS**
  General Ledger, Inventory, Accounts Payable, Accounts Receivable, Word Processing Systems.
- **GENERAL APPLICATIONS**
  Northstar Macro Assembler $65
  Northstar Mailing Label Program $45
- **STOCK MARKET PACKAGE** (Unique) Makes Makes Ticker-Tape Obsolete.
  Send $2.00 for descriptive brochure on Stock Market Package.

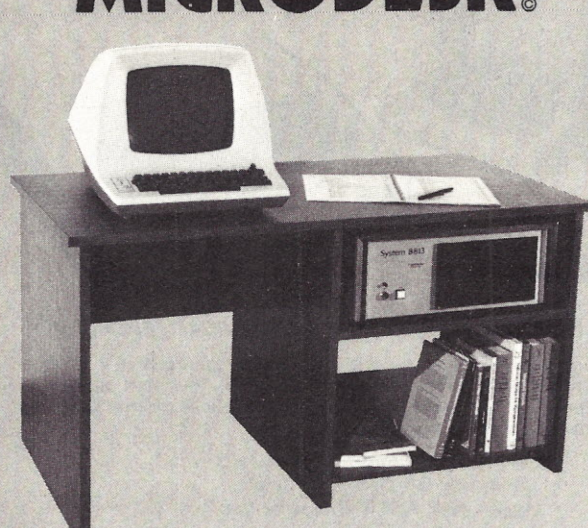
BYTE SHOP B14

THE AFFORDABLE COMPUTER STORE T.M.

130 East 40th Street	**2721 Hempstead Turnpike**
New York, NY 10016	**Levittown, NY 11756**
(212) 889-4204	**(516) 731-8116**
(Corner Lexington Avenue)	**(Just East of Wantagh Pkwy)**

Tues. thru Fri. 11 to 7 . . . . 12 to 8
Saturday 10 to 5. . . . . 10 to 5

MICRODESK©



The perfect home for a personal or business computer

48" x 24" x 26", Shelves 23" x 23", One Adjustable
MICRODESK $96.50 113 lbs. F.O.B. Wichita

COMPUTER SYSTEMS DESIGN
1611 E. Central Wichita, Kansas 67214
DEALER INQUIRIES INVITED

DEALER DIRECTORY

Los Angeles CA

Featuring: PolyMorphic, North Star, Imsai, Cromemco, Extensys, Speechlab products and Poly-88 Users Group software exchange. All products 10-20% off list. We won't be undersold! **A-A-A-A Discount Computer How's, 1477 Barrington, Suite 17, Los Angeles CA 90025, 477-8478.**

Palo Alto CA

Systems available for immediate delivery: word processing; multiprogramming, multiuser work in process; business; medical/dental billing and accounts receivable. Software and hardware guaranteed. **Byte of Palo Alto, 2233 El Camino Real, Palo Alto CA 94306, 327-8080.**

Chicago IL

Staffed by computer professionals. Largest selection of books and magazines in the Chicago area. Specialists in CP/M, Apple Computer and Alpha microsystems. **Data Domain of Schaumburg, Plaza de las Flores, 1612 E. Algonquin Rd., Schaumburg IL 60195, 397-8700.**

Grand Rapids MI

Full-line microcomputer store. Ohio Scientific—Equinox—PolyMorphic Systems—Digital Systems—Godbout —Dynabyte —Thinker Toys—Meca—North Star. **Micro Computer World, 313 Michigan St. N.E., Grand Rapids MI 49503, 451-8972.**

Royal Oak MI

A full-time sales staff combined with a full-time software staff and a full-time technical staff means full support for your computer. **The Computer Mart, 1800 W. 14 Mile Rd., Royal Oak MI 48073, 576-0900.**

Omaha NE

Business systems, programming, and maintenance. Hobbyists welcomed. Satisfied customers our goal. Open seven days/nights. **Byte Shop of Omaha, 8523 Park Dr., Omaha NE 68127, 339-7350.**

Nashua NH

Small business computer systems: Imsai VDP-80, PolyMorphic Systems 8813, Seals PUP-1, Algorithmics Word Processing System. General ledger, A/R, A/P, invoicing, payroll. Specialized microprocessor applications. **Microcomputers, Inc., 36 Otterson St., Nashua NH 03060, 889-1646.**

Syracuse NY

We stock: Apple, Commodore PET, KIM, North Star, Cromemco, Imsai, TDL, Vector Graphic, Heuristics, Lear Siegler, Netronics ELF, books, magazines, parts and supplies. **Computer Shop of Syracuse, Inc., 3470 Erie Blvd. East, DeWitt NY 13214, 446-1284.**

Tell them you saw their name in Kilobaud

Newton NC

Turnkey microcomputer systems for professional, business and industrial applications, featuring Imsai (and the fabulous VDP-80). Custom software, expert service and sound advice from DP specialists. **Microcomputer Services, Rte. 1, Box 337, Newton NC 28658, 462-1401.**

York PA

Personal and small-business systems. Components and peripherals. Books, magazines, information. Hobbyists Welcome. ½ Block from Business 83 at Iron Bridge. **G.Y.C. Co., 51 Hamilton Ave., York PA 17404, 854-0481.**

Houston TX

Bare boards, kits, small systems, multiuser systems, custom hardware design, software for business or hobby, nationwide consulting available. Please call. **Houston Computer Mart, 8029 Gulf Freeway, Houston TX 77017, 649-4188.**

Dealers

Interested in being listed in *Kilobaud's* (prepaid) Dealer Directory? You should be! (Just wait until you hear the nominal cost! **Contact Heidi! 603-924-3873.**

Richland WA

Apple, North Star, Ohio Scientific. All products listed in Byte Shopper, classes, literature, electronic parts, repair services. Custom design/consulting in both hardware and software. **Ye Olde Computer Shoppe, inc., 1301 George Washington Way, Richland WA 99352, 946-3330.**

Spokane WA

Clearance sale on Imsai. Discounts to 60%; to 33% on mainframes. One V DP-80 with IMDOS, CBASIC $1400 below retail. Distributor for ADDS terminals. **The Computer Shop, S. 107 Wall St., Spokane WA 99204, 456-0311.**

Madison WI

Mainframes by Commodore, Processor Tech., Cromemco, Imsai, TDL. Printers by Teletype, NEC, Texas Instruments, Diablo. Terminals by Microterm, Intertube. Micropolis disk drives. Service and software. **The Madison Computer Store, 1863 Monroe St., Madison WI 53711, 255-5552.**

Milwaukee WI

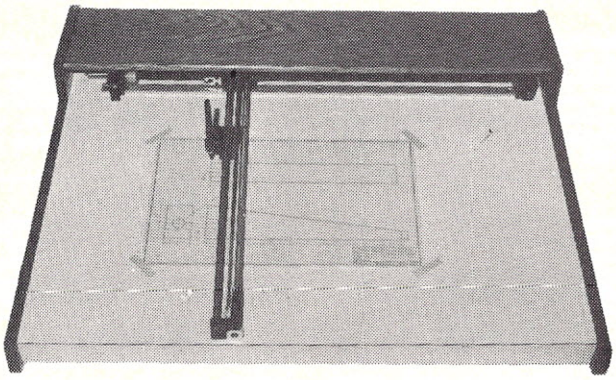
We carry the following manufacturers' products: Byte, Imsai, Cromemco, Hazeltine, Centronics, Sanyo, Dynabyte, North Star, Micropolis. We are also a PET dealer. Excellent service. **Byte Shop of Milwaukee, 6019 West Layton Ave., Greenfield WI 53220, 281-7004.**

Is your city listed?

INTRODUCING THE

X - Y PLOTTER UNIT

BY SYLVANHILLS



. . . To interface to your business or hobby computer.

FEATURES :

- *LOW COST*
- *INCLUDES POWER SUPPLY AND INTERFACE*
- *RUGGED AND RELIABLE*

DRAWING AREA

11 X 17 inch.	$ 1049.
17 X 22 inch	$ 1249.

AND

FOR THE DO-IT-YOURSELFER:

X - Y PLOTTER KITS

ASSEMBLED
CONSOLE AVAILABLE
POWER SUPPLY AVAILABLE

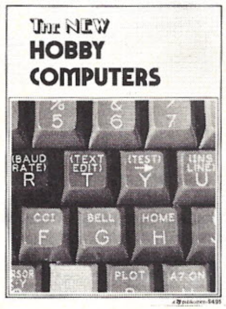
11 X 17 inch	$795.
Matching Console	$110.
17 X 22 inch	$950.
Matching Console	$130.

REQUEST: FREE BROCHURE
OR
OWNERS' MANUAL $5
OUTSIDE US AND CANADA ADD $3
FOR AIR POSTAGE

Sylvanhills Laboratory, Inc.

P.O. BOX 646 PITTSBURG, KANSAS 66762 316 - 231 - 4440

SEND FOR THESE GREAT BOOKS TODAY!



• **HOBBY COMPUTERS ARE HERE!** If you (or a friend) want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fundamentals and explains the circuits, the basics of programming, along with a couple of TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid for newcomers. $4.95

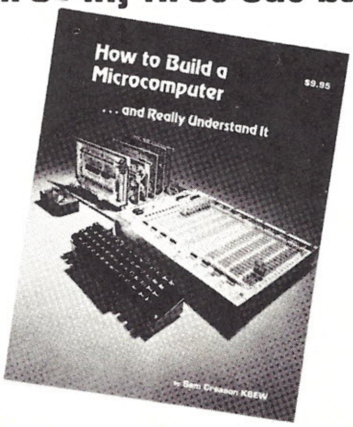
• **THE NEW HOBBY COMPUTERS** This book takes it from where "HOBBY COMPUTERS ARE HERE!" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards, a Baudot monitor/editor system, an audible logic probe for finding those tough problems, a ham's computer, a computer QSO machine . . . and much, much more! Everything of interest is there in one volume. Don't miss this tremendous value! Only $4.95

Have your credit card handy and call toll-free 800-258-5473, or use the card in the back of this magazine and send your check to:

KILOBAUD Book Dept. • Peterborough, NH 03458

SPECIAL: PRE-PUBLICATION OFFER

(orders will be filled on a first-in, first-out basis)



• **HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT**—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. Sam Creason's book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing, and debugging of a complete microcomputer system. This book is must reading for anyone desiring a true understanding of small computer systems. $9.95.*

Have your credit card handy and call toll-free 800-258-5473, or use the card in the back of this magazine and send your check to:

KILOBAUD Book Dept. • Peterborough NH 03458

Kilobaud Classified

Kilobaud classified advertisements are intended for use by those individuals desiring to buy, sell, or trade used computer equipment or software. No commercial ads are accepted.
Two sizes of ads are available. The $5 box allows five lines of about 22 characters each, including spaces and punctuation. The $10 box provides ten lines of type — again, each line is about 22 characters. Minimize capital letters, as they use twice the space of small characters. Payment is required in advance with ad copy. We cannot bill, or accept credit. Oversize ads are not accepted. Each subscriber is limited to two (2) identical ads in any given issue.
Advertising text and payment must reach us 60 days in advance of publication. For example, advertising copy for the March issue (mailed in February) must be in our hands on January 1. The publisher reserves the right to refuse a questionable or not applicable advertisement. Mail advertisements to: KILOBAUD CLASSIFIED, Kilobaud, Peterborough, NH 03458. Do not include any other material with your ad, as it may be delayed.

TRS-80 Monthly Newsletter. For information write to: Howard Y. Gosman, Box 149, New City NY 10956.

Home accounting system written in BASIC for North Star disk or others. Maintains paid and projected expenses by minor and major categories. $9.50 or $14.50 with disk. Dieter Kaetel, 7201-87 S.E., Mercer Island WA 98040. (206) 232-1513.

TRS-80 Level I 4K—5 pop. games. Orig. —graphics & score—cassette, $7. C. Zalnerunas, 3034 W. Columbus Ave., Chicago IL 60652.

Extensys 48K dynamic RAM board; Expandable to 64K; S-100-bus compatible; New. $1100 or best offer. W. Evraiff, 1976 Ticonderoga Dr., San Mateo CA 94402.

TRS-80 Sci-Fi Games (4K, LI): Galactic blockade runner—a sophisticated space war game. $9.95 for cassette and 12 pg. manual; SASE for game list. Tim Quinlan, 219 Washington Ave., Chelsea MA 02150.

Computer-Aided Handicapping. Program computes win probabilities of entries in thoroughbred horse races. Based on stat. analysis. Guide incl. Written in BASIC; 4K. $40 or write for specs: D. Cammert, Box 3844, Thousand Oaks CA 91360.

TRS-80 in BASIC I, 4K min., sophisticated graphic games your computer can play. Hangman with TRS selecting word and drawing picture as you guess letters. 3-D Tic-Tac-Toe. Very hard to beat. Computer picks its moves in 10 seconds. Boards are displayed. $10 on tape w/doc. J. Campbell, 6927 Atoll Ave., N. Hollywood CA 91605.

Logical Services programming course. $30 postpaid. Don Ketchum, 313 Van Ness Ave., Upland CA 91786.

PET Software: Try these graphic programs. Blackjack, Biorhythm, Amortization, Depthcharge, Tic-Tac-Toe (3-D), Multiplication, Hangman. All ready to load for your 8K PET. Only $6 each or 3 for $15. A. L., PO Box 7211, Grand Rapids MI 49510.

For Sale: H9 TV term. Assembled. $450 includes shipping & books. M.O. to: Steve Vickery, PO Box 1548, Panama City FL 32401.

Poly & TRS-80 Software—New, original, large selection. Send for a complete list. Ted Carter, 902 Pinecrest, Richardson TX 75080. (214) 235-0915.

Regulated power supplies, tested working 5 V @ 25 Amp, $95, freight collect, UPS. A. Khan, 2355 Burch Cir., N.E., Atlanta GA 30319. (404) 325-0089.

North Star Software Specials: Investment Analysis Package: $25. Home/Family Package: $25. Game Pack: $15. All these disks are full and just packed with programs! Order now, or write for complete list. Herbert Schildt, 1007 N. Division, Urbana IL 61801.

Teletype, ASR33 with paper tape. Inexpensive ASCII small systems terminal. Modern, compact styling. Excellent. Several available, $495. (804) 272-8403, evenings.

TRS-80 Owners: ROM Check Tape, $3.95. Tests for errors known to be in Level 1 ROMs. Evaluate your system! Also: Programmer's Reference Card, $1.25. A thorough description of Level 1 BASIC in pocket size! E. J. Bivins, 1846 W. Broadway, Anaheim CA 92804

APPLE II Software—Word Processor, $50 —Universal Database, $50—Inventory/ Point of Sales/Sort/Reorder, $125. For lit. send $1 to Darrell Smith, 17638 157th Ave., S.E., Renton WA 98055.

Variety of programs on cassette for Radio Shack TRS-80. Details from Dr. S. Harter, Box 17222, Tampa FL 33682.

For Sale: Magnetic Tape Cleaner. Virginia Panel Corp., Model 24. Mechanically cleans, IBM-compatible snap-lock hubs, ½-inch tape. Price new, $2395. Sacrifice, $925. Don Magill, 2909 Stonybrook Dr., Bowie MD 20715.

Sell: Elf II, 4K, giant board, pwr supply, $200. W. Cope, 302 Anderson St., Durham NC 28805. (919) 684-0893.

RCA COSMAC VIP Owners: Backgammon against computer. Fantastic! Also: Vegas Blackjack & Craps; Pinball; Star Wars AND available soon, Checkers! SVS, North Hollywood CA 91601.

TRS-80 with all peripheral equipment. Will sell complete or separate. Norman Frank, 1117 Conway, Mission TX 78572.

PET Users. Cassette and instructions to write and read directly to and from memory for fast-execution machine-language programs. $4.95 postpaid. Don Ketchum, 313 Van Ness Ave., Upland CA 91786.

4K TRS-80 graphics games: Othello, Mastermind, Racer, Wumpus, Reverse, Escape, Guesser, Battleship, Life, Anims, Lunar, Star Trek. Fully documented. 2 for $5 or all for $15. Bargains: 704 Washington St., Ogdensburg NY 13669.

PET User Group. To share and exchange applications, programs and hardware expansion techniques. First year membership—$5 for 6 issues of PET User Notes. Gene Beals, PO Box 371, Montgomeryville PA 18936.

New Ithaca Audio Kits: 8K RAM—$125; Z-80 CPU w/ EPROM option—$135; 16K (2708) EPROM—$55, in orig. boxes. 7 & 8 pos. DIP switch—$1.50; 16/14 pin IC sockets—$16/14 per 100. W. Imhauser, 8797 Ottawa Riv., Fountain Vly CA 92708. (714) 968-0285 aft. 7 PM.

TRS-80, 4K min. BASIC Level I. Unique horse race program on cassette tape. Can vary length of track, change pace and change speed of horses. $10 on tape w/doc. To: EMFW, Box 438, Oakhurst NJ 07755.

Wanted: Game programs for TRS-80. Royalties. Send Description. E. J. Bivins, 1846 W. Broadway, Anaheim CA 92804.

All prime, 100% guaranteed, in tubes. CD4030—25/$7.50, CA3079—25/$12, 74S112—25/$10, 74116—15/$15. A. Khan, 2355 Burch Cir. N.E., Atlanta GA 30319.

Microcomputers in Psychiatry: Bimonthly newsletter for sharing ideas and experiences. Send $6 to Marc Schwartz M.D., 26 Trumbull St., New Haven CT 06511.

TRS-80 software on cassette or diskette. Lowest cost. Free newsletter. Poi Pow, 96 Dothan St., Arlington MA 02174.

PET users need info? Getting started with your PET workbook, $4. Workbook has helpful info, sample programs, exercises and fundamental features of PET BASIC. Send SASE for details. D. Smith, Box 921, Los Alamos NM 87544.

GE opto-isolators H11A2, all prime. 100% guaranteed. With spec. 2/$1, 100/$40. A. Khan, 2355 Burch Cir. N.E., Atlanta GA 30319.

CIRCLE CHESS instruction book by Alphonso, used in Circle Chess software development. $5, Stanonis, Box 63, Des Plaines IL 60017.

PET Owners: Send for our free catalog of PET software. Custom programs available. C. M. Stuart, 5115 Menefee Dr., Dallas TX 75227.

TRS-80 Level I prgs. on cassette. 4K: Lunar/ Martian Lander, ($8); 16K: Star Trek w/ graphics, ($10); Oklahoma-Texas Football, ($12); Carrier Battle, ($6). R. Fitzgerald, 3338 Sheila Ln. 292, Dallas TX 75220.

We had replacement panels made for our Altair 8800s. They're blue against a white background and really look nice! We'll sell the extras for $16 each. D. L. Tibbetts, 520 Schoolhouse Ln., Willow Grove PA 19090.

Blackjack Simulator for the TRS-80 in BASIC I. Program allows you, with a minimum knowledge of BASIC, to write routines to test your 'system' for beating BJ. Simulator calls your routines at hit, ins. and bet time. Plays thousands of hands and records statistics on tape. Pkg. incl. analysis program to display statistics. Progs. on tape w/doc., $20. Send to: E. V. Reider, Box 6153, Syracuse NY 13217.

Southern Connecticut: Anyone using Tarbell disk system, please contact me for mutual system support. Also need help in maintenance, fine points of CP/M; will pay! James Van Pelt, 25 Sagamore Cove, Branford CT 06405 (New Haven area).

Contest!

Take a look at the reader-service card in the back of *Kilobaud* and you'll notice a space to vote for what in your opinion is the best article in the issue. As votes come in, they are tabulated, and the author of the article receiving the most votes receives a check for $100.

Then, we draw a card from among all those submitted; the person whose card is chosen gets his choice of a book from the KB Book Nook.

Additionally—since our advertisers depend on reader feedback, all readers who submit cards are eligible to be entered in a monthly drawing for a lifetime subscription to *Kilobaud*.

So, it pays to write for *Kilobaud*, to read it and to send in reader-service cards.

By the way—the article winner for May 1978 is Ralph Wells for "PET's First Report Card."

The Book Nook winner is Richard Shiffman of Los Angeles CA.

Winner of a lifetime subscription is listed on page 5.

Our congratulations to all.

A Reminder

If you write an article for Kilobaud, and if your manuscript includes program listings and printouts, please submit them in *camera-ready* condition. That is: If at all possible, use a good-quality printer that produces well-defined letters; and use a new or relatively new ribbon. Please use white paper.

If access to a printer is absolutely out of the question, then please carefully type the listings (single spaced) on white paper using a fresh ribbon.

In the long run, this makes things easier for everyone. We will greatly appreciate your assistance.

Huge Discounts!

HOBBY WORLD ELECTRONICS

Savings up to 70% on major brand IC parts and computer kits. For complete IC listings write for our catalog.

KITS

WAMECO S-100 P.C. Boards
- 8K RAM 28.00
- 8080 CPU 28.00
- 12-Slot Mother Board 33.00

ITHACA AUDIO S100 P.C. Boards
- 8K RAM 28.00
- Z-80 CPU 28.00
- 2708/2716 EPROM Boards (New) 28.00
- W/W Prototype Board (New) 25.00

SOLID STATE MUSIC S-100 Kits & Bare Boards
- MB6A 8K Kit 129.95
- STATIC RAM Bare Board 25.95
- MB7 16K STATIC RAM Kit 435.00
- Bare Board 25.95
- MB8 8K/16 K EPROM USES 2708's Kit Less EPROMs 75.95

SPECIAL

HUH ELECTRONICS 6 ten minute data cassettes: $6.50 (New)
5¼" MINIDISCETTES; Soft Sector, 10, 16, each $4.65 (New)
8" FLEXIBLE DISCS; Hard or Soft Sector, each $5.33 (New)

KITS

SOLID STATE MUSIC
VB1B VIDEO INTERFACE
- Kit 129.95
- Bare Board 25.95

RF Modulator TV-1 Converts Video Output to RF For TV antenna terminals, complete Kit 8.95

NEW FROM CYBERCOM DIV OF SSM
- MB8A—2708 16K EPROM with Vector Jump Kit 85.00
- OB-1 Vector Jump & Prototype Kit 49.00
- Board 25.95
- IO4 2+2 I/O Kit 139.95

SYNTHESIZER SB-1 MUSIC Kit with Software 145.00
MT-1 15-Slot Mother Board 39.95
XB-1 EXTENDER BOARD Bare Board 8.99
SSM 8080 MONITOR VI ON 2-2708 47.00
ON 8-1702A 47.00

BOOKS

The Basic Workbook	$ 5.50
Programming Proverbs	6.95
Discovering BASIC	6.85
COBOL with Style	6.95
Advanced BASIC	7.95
Standard Dictionary of Computers & Information Processing	16.95
Game Playing with Computers	16.95
Game Playing with BASIC	6.95
Minicomputers	13.95
Microprocessors	10.95
Digital Experiments	8.95
Digital Signal Analysis	19.95
Digital Troubleshooting	9.95
110 CMOS Digital IC Projects	5.95
400 Ideas for Design, Volume 2	13.75
Analysis and Design of Digital Circuits and Computer Systems	22.50
Telephone Accessories You Can Build	4.95
Basic Electronic Switching for Telephone Systems	6.95
Basic Carrier Telephony	6.85

(Continued)

8080 Machine Languages Programming	6.96
Robots At Your Doorstep	6.95

BEGINNERS BOOKS

Beginning BASIC	9.95
Home Computer Beginners Glossary and Guide	6.95
Basic BASIC	$ 8.95
Introduction to BASIC	8.95
Home Computers: 210 Questions and Answers Volume 2: Software	6.95
Microcomputer Dictionary and Guide	19.95
Microprocessor Basics	10.95
Home Computers: 210 Questions and Answers Volume 1: Hardware	7.95
Understanding Integrated Circuits	4.95
Semiconductor Circuit Elements	6.95
Fundamentals and Applications of Digital Logic Circuits	7.95

74L$00 TTL

74L$00	.21	74L$138	.70
74L$02	.21	74L$139	.70
74L$04	.24	74L$151	.65
74L$08	.21	74L$153	.66
74L$10	.21	74L$154	1.00
74L$14	.85	74L$157	.62
74L$20	.23	74L$160	.82
74L$21	.23	74L$161	.82
74L$22	.23	74L$162	.82
74L$30	.23	74L$163	.82
74L$32	.30	74L$164	.98
74L$37	.31	74L$174	.75
74L$38	.31	74L$175	.79
74L$42	.60	74L$190	.90
74L$47	.75	74L$191	.90
74L$48	.72	74L$192	.90
74L$73	.35	74L$196	.80
74L$74	.35	74L$197	.80
74L$75	.53	74L$221	1.06
74L$76	.37	74L$257	.71
74L$86	.36	74L$258	.70
74L$90	.52	74L$266	.26
74L$92	.52	74L$283	.72
74L$93	.52	74L$365	.55
74L$109	.36	74L$366	.55
74L$112	.36	74L$367	.55
74L$113	.36	74L$368	.55
74L$114	.36	74L$386	.39
74L$125	.46	81L$95	.77
74L$126	.46	81L$96	.77
74L$132	.75	81L$97	.77
		81L$98	.77

SPECIAL

(New) 25 Pin—D Subminiature Solder Tail
Male 3.00
Female 3.50

100 PIN EDGE CONNECTOR, Imsai spacing 3.50 (New)

SUPPORT DEVICES

6820	8.00
6850	8.00
8212	3.45
8214	8.00
8216	3.75
8224	3.50
8228	6.25
8226	3.85
8238	7.95

MEMORY

1702A	3.75
2708	11.50
2716 TI	22.50
21L02 450ns	1.25
21L02 250ns	1.60
2114	8.50

MICROPROCESSOR

8080A	11.50
Z-80	24.95
Z-80A	34.95
6800	16.50

CMOS

4001	.16	4025	.16
4002	.16	4028	.74
4006	.99	4030	.22
4007	.16	4035	.98
4008	.79	4040	.99
4009	.37	4049	.34
4010	.37	4050	.34
4011	.16	4051	.89
4012	.16	4052	1.15
4013	.29	4053	.89
4014	.74	4066	.59
4015	.74	4071	.19
4016	.29	4081	.23
4017	.79	4082	.23
4018	.79	4507	.39
4019	.23	4510	1.05
4020	.84		
4021	.89		
4022	.89		
4023	.16		
4024	.67		

74XX

7400	.11	7482	.50
7401	.13	7483	.62
7402	.13	7485	.75
7403	.13	7486	.26
7404	.15	7489	1.75
7405	.13	7490	.40
7406	.16	7491	.51
7407	.23	7492	.40
7408	.17	7493	.40
7409	.17	7494	.60
7410	.13	7495	.60
7411	.18	7496	.60
7412	.13	74107	.28
7413	.25	74109	.22
7414	.61	74121	.29
7416	.24	74122	.38
7417	.22	74123	.48
7420	.13	74132	.65
7423	.25	74141	.70
7425	.25	74145	.65
7426	.22	74150	.88
7427	.17	74151	.61
7430	.13	74153	.61
7432	.23	74154	.95
7437	.21	74157	.55
7438	.21	74161	.55
7439	.25	74163	.55
7440	.13	74164	.85
7441	.70	74165	.90
7442	.37	74170	1.69
7443	.59	74173	1.10
7444	.59	74174	.85
7445	.65	74175	.75
7446	.62	74176	.69
7447	.59	74177	.70
7448	.60	74180	.65
7450	.13	74181	1.75
7451	.13	74190	.95
7453	.13	74191	.95
7454	.13	74192	.79
7460	.14	74193	.80
7470	.26	74195	.49
7472	.21	74221	.86
7473	.21	74251	1.00
7474	.27	74365	.62
7475	.45	74366	.62
7476	.28	74367	.62
7480	.31	74368	.62
7481	.95		

H25

5716 W. Manchester Ave.
Suite #5
Los Angeles, CA 90045
TELEPHONE ORDERS: Call (213) 641-4200

master charge THE INTERBANK CARD

BANKAMERICARD VISA

☐ Cash
☐ COD

Charge My
☐ M.C.
☐ BAC (VISA)

Exp. Date:

Sig.

☐ Send your complete catalog, quickly.
☐ Please send me the following items I have listed below:

Qty.	Stock No.	Price
	Postage/Handling	$1.50

Name

Address

City State Zip

Satisfaction 100% Guaranteed

California Residents Add 6% Sales Tax

Note: Minimum Order $10.00, 5% Discount over $100.00 on I.C.'s only.

NEW 1978 IC MASTER

- Over 40,000 IC's listed.
- Over 2,000 pages.
- Updated every 90 days.

Retail Value **$55.00** Your Price **$39.99**

Check catalogue for more listings on linears, voltage regulators, resistor books, IC sockets and more.

California Industrial

Post Office Box 3097 K • Torrance, California 90503

Hazeltine 1400

cost effective

CRT TERMINAL

$659 plus shipping

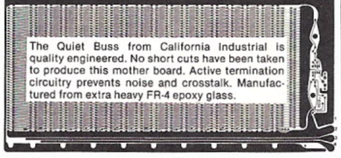
The Hazeltine 1400 Vidio Display Terminal is designed to optimize interactive real-time operations. The interface is capable of either local or remote connection through an EIA RS232-C interface at baud rates that are switch selectable up to 9600 baud.

- All 128 ASCII Codes
- 64 Displayable Characters
- 24 Lines; 12 inch Screen
- 80 Characters per Line
- Self Diagnostic Test

Quantity pricing upon request.

S-100 Mother Board

Quiet Buss™ $29.95



The Quiet Buss from California Industrial is quality engineered. No short cuts have been taken to produce this mother board. Active termination circuitry prevents noise and crosstalk. Manufactured from extra heavy FR-4 epoxy glass.

18 slot IMSAI

TELETYPE® MODEL 43

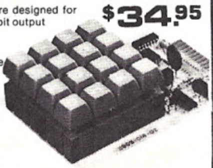
New from Teletype, the Model 43 is capable of printing 132 ASCII characters per line. Send and receive data at 10 or 30 char. per second. Keyboard generates all 128 ASCII code combinations. RS-232 interface, same as the popular Model 33. Data sheet sent upon request. Manufacturer suggested price $1377.00.

IMMEDIATE DELIVERY $1219

TTL model with NOVATION brand Acoustic Modem. $1419

HEXADECIMAL KEYBOARD

$34.95



Maxi-Switch hexadecimal keyboards are designed for microcomputer systems that require 4-bit output in standard hex code.

Each assembly consists of 16 hermetically sealed reed switches and TTL "one shot" debounce circuitry.

Reliable low friction acetal resin plungers are credited for the smooth operation and long life of this premium keyboard.

Requires single +5 volt supply.

MAXI-SWITCH NEW ASCII KEYBOARD

Fully Assembled **$64.97**

Definitely the best small system keyboard that we have seen. Maxi-Switch has incorporated all the important keyboard features at a reasonable price. Full 128 ASCII functions, "N" key rollover, automatic repeats, user designated special function keys, escape, control & lots of others. Data sheet upon request.

CONNECTORS

your choice DB25P male plug & hood or DB25S female **$3.95**

Qty.	fe.	male	hd.
10	3.45	2.45	1.15
25	3.15	2.25	1.05
100	2.85	1.90	.95
500	2.25	1.60	.85
1K	1.97	1.37	.73

Edge Connectors

100 PIN IMSAI/ALTAIR

S-100 • GOLD PLATED • .125" CENTERS

Altair .140 row, soldertail..... $5.98 3/$16.50

Imsai .250 row, soldertail..... $4.98 3/$11.95

3 Level Wire Wrap .250 row... $4.98 3/$13.00

SPECIALS

W/W same as above without ears $3.50 3/$10

72 (dual 36) W/W .156" centers... $2.50 3/$6

SPERRY UNIVAC KEYBOARD

$24.88

The famous Sperry Univac 1710 Hollerith keyboard assembly is now available from California Industrial for only $24.88. The ideal computer input device for accountants and mathematicians. The numeric keys are placed on the lower three rows to resemble a ten key adding machine. This format allows one handed numeric data entry. Original cost was $385. Used but guaranteed in excellent condition. Complete with documentation.

FROM ATARI COLOR TELEVISION R.F. MODULATOR

$13.95

The Atari R.F. Modulator allows computer data to be displayed directly upon your existing television system. This unit converts the signal from the Apple II and other video sources into television frequencies. Operates from single 5 volt supply. Complete with metal case, mating R.F. connector and 15 feet of coax cable. Schematics and instructions included.

Assembled

FREE

MANUAL GRAPHITE DISPLAY GENERATOR

Modern technology has pioneered the development of this unique character printer. Our Manual Graphite Display Generator has the capability of producing the full upper and lower case ASCII set. Self-contained cursor assembly allows the operator to eliminate erroneously entered information. Each unit is manufactured to strict tolerances as prescribed by standards set forth by California Industrial. One free with every order.

Scotch® BRAND

$4.98 10 for $45.

DISKETTES

8 inch Soft (IBM)
8 inch 32 sector
Mini Soft sec.
Mini 10 sector
Mini 16 sector

Certified Digital **CASSETTES**

Won't drop a BIT!

$5.50

CALIFORNIA INDUSTRIAL is an Authorized Dealer of Scotch Brand Data Products

JOYSTICK $4.50

3 for $10.00

This joystick feature four 100K potentiometers, that vary resistance proportional to the angle of the stick. Perfect for television games, quad stereo and radio controlled aircraft.

450ns 2708

$9.95

8K UV Erasable MEMORY

Potter & Brumfield REED RELAY

5 volt coil, pulls 3.5v.

$1.19 ea.

	10	100	1K
SPST	.97	.79	.65

S-100 PROTOTYPE BOARD

$19.98

GP100-Maximum design versatility along with standard address decoding and buffering for S100 systems. Room for 32 uncommitted 16 pin IC's, 5 bus buffer & decoding chips, 1 DIP address select switch, a 5 volt regulator and more.

WW100-Wire wrap breadboard, similar to the GP100. Allows wire wrap of all sizes of sockets in any sizes of sockets in any combination. An extra regulator position for multiple voltage applications.

21L02

LOW POWER 450 nS

SUPER BUY! $1.19

Lowest Price Anywhere

Our low power static RAMs are factory prime. Purchased, on contract, directly from one of California's leading semiconductor manufacturers.

Access time guaranteed faster than 450nS. **Minimum purchase 32 pieces.**

Sorry, credit cards can not be accepted on 21L02's.

7400	.13	7442	.59
7401	.19	7443	.79
7402	.19	7445	.89
7403	.19	7446	.99
7403	.19	7447	.99
7404	.19	7448	.99
7405	.19	7450	.25
7406	.19	7451	.25
7407	.25	7453	.25
7408	.25	7454	.25
7409	.25	7460	.25
7410	.19	7470	.25
7411	.25	7472	.39
7412	.35	7473	.39
7413	.49	7474	.35
7414	.79	7475	.49
7416	.39	7476	.39
7417	.39	7479	3.99
7420	.19	7480	.79
7422	.49	7482	.99
7423	.39	7483	.99
7425	.39	7485	.99
7426	.39	7486	.49
7427	.39	7488	3.40
7428	.49	7489	2.79
7429	.39	7490	.49
7430	.25	7491	.99
7433	.39	7492	.49
7437	.39	7493	.49
7438	.39	7494	.79
7439	.39	7495	.79
7440	.39		

7496	.79
7497	3.99
74100	1.19
74107	.39
74109	.49
74110	1.79
74116	1.99
74120	1.79
74121	.39
74122	.39
74123	.69
74125	.59
74126	.59
74128	.49
74132	.99
74136	.89
74141	.99
74145	.99
74147	2.49
74148	1.99
74150	1.19
74151	.99
74153	.89
74154	.99
74155	.99
74156	1.29
74157	.99
74159	2.99
74160	1.19
74161	.99
74162	1.49
74163	.99
74166	1.19
74167	4.99
74170	2.49
74173	1.49
74174	1.19
74175	.99
74176	.99
74177	.99
74192	.99
74193	.99

CMOS

4001	.25	4033	1.99
4002	.25	4035	1.69
4006	1.99	4040	1.99
4007	.25	4041	1.49
4008	1.49	4042	1.49
4009	.69	4043	1.49
4010	.69	4044	1.49
4011	.25	4046	2.49
4012	.25	4047	2.49
4013	.49	4049	.79
4014	1.49	4050	.79
4015	1.39	4051	1.99
4016	.69		
4017	1.29		
4018	1.69		
4019	1.79		
4020	1.39		
4021	1.49		
4022	1.25		
4023	.25		
4024	1.19		
4025	.25		
4027	.69		
4028	1.25		
4029	1.99		
4030	.69		
4032	.49		

CPU's

8080A	14.95
6800	24.95
Z80	39.95

MEMORY

1702A	4.95
82s23	2.95
82s123	2.95
2102	1.79
2102-1	1.89
21L02	1.19
250nS	1.49

CLOCK's

5314	2.95
5316	4.95
5375	3.95

LM741 $ SOLD OUT

301H	.39	350N	.99
301CN	.39	351CN	.65
302H	1.29	370H	1.29
304H	1.29	370N	1.29
305H	.99	373N	3.19
307H	.49	377N	3.99
307CN	.39	380N	1.39
308H	.99	381N	1.79
308CN	.99	382N	1.79
309H	1.09	NE555v	.49
309K	.99	NE556	1.29
310H	1.19	NE565H	1.49
310CN	1.19	NE565N	1.79
311H	.99	NE566N	1.25
311N	.99	703CN	.45
312H	1.99	709H	.39
318H	1.79	709N	.39
318CN	1.49	710N	.79
319CN	1.29	711H	.39
320K-5		711N	.39
320K-12		723H	.55
320T-5		723N	.55
320T-12		725H	3.49
		733H	1.49
		733N	.99
		739N	1.19
		741N	.09
		747N	.79
340K-5		748H	.39
		748N	.39
340K-12		1414N	1.75
		1458	.69
340T-5			
340T-12			

320K-5 through 340T-12: 1.19

UART AY 5-1013A **$4.98**

SPECIAL

GENERAL INSTRUMENT ASCII Keyboard Encoder AY 5-3600 Prime but house marked only $4.95

Digital Cassette Drive

COMPUTER CONTROLED

$79.50

This precision I/O assembly features remote software controlled search capabilities. Two independent capstan drive motors allow the computer to control direction and speed of the transport.

The assembly consists of a Raymond cassette transport, chassis, motherboard and three edge cards: read/write, capstan drive & control card.

Current replacement valued at over $700.00. Schematics and complete documentation included. USED, but in excellent condition.

Thumbwheel switch

Ten position **BCD**

$1.39 ea.

10	50
$1.19	.89

MINIATURE SWITCHES

your choice

$.98

10	50	100	1k
$.88	.81	.73	.66

SPDT Miniature Toggles

7101 C&K ON-NONE-ON
7107 jbt ON-OFF(mnt.ON)
7108 CK ON-(moment. ON)
Rocker JBT ______ DPDT
Rotary 3P-4-Pos.
Rotary 3P-6-Pos.
Push B (N.O.) $.39ea. 4/$1

CAPACITORS

ELECTROLYTICS

	ea.	10	50
80,000/10v.	3.95	3.49	2.95
4500/50v.	$1.49	1.35	1.19
1000/15v axial	$.55	49	.45
.1 disc	$.12	.09	.07
.01 disc	.06	.05	.04

DIP Switch

$1.49 ea.

10	25	100	1K
$1.29	1.15	.97	.85

specify 7 or 8 pos.

SPST CTS

50 Conductor RIBBON WIRE

$.69 Ft.

TWISTED PAIR

Power Adapter

6 vdc, 140mA $1.39
7 vdc, 1.4 A. 5.50
9 vdc, 15mA. 1.19
10 vAc, 300mA. 1.95

RELAYS

SPDT MINIATURE

$1.19 ea.

10	25	100
$1.15	1.04	.89

Coil 12 Volt dc.
7 Amp Contacts
P. C. Board Mount

Transistors

	ea.	10	50	100
2N2222A	.20	.18	.16	.15
2N3055	.69	.65	.59	.55
MJ3055	.79	.75	.69	.65
2N3772	1.59	1.49	1.39	1.29
2N3904	.15	.11	.09	.07
2N3906	.15	.11	.09	.07

Diodes

		10	25	100
1N4002	100v.	.08	06	.05
1N4005	600v.	.10	.08	.07
1N4148	signal	.07	.05	.04

jumbo red **LED's**

ea.	10	25	100
$15	.13	.11	.09

BINDING POSTS 5-WAY

3 for $1.19

20	100
$.35	.29

9 foot

$1.49

Heavy duty grounded power cord and mating chassis connectors.

SOLDERLESS TERMINALS

INSULATED 20 for $.98

Specify: 22-18; 16-14

100	500	1k
4.50	20.	35.

PANASONIC

$.98

Nicad "A A" 450mA.

DISCOUNT Wire Wrap Center

IC SOCKETS

pin	wire wrap ea.	25	50	low profile ea.	25	50
8				17¢	16	15
14	37¢	36	35	18	17	16
16	38	37	36	19	18	17
24	99	93	85	36	35	34
40	1.69	1.55	1.39	63	60	58

KYNAR WIRE WRAP

50ft. $.98

500	1,000	11,000
$9.	$15.	$105.

Battery wire wrapping tool ~~$34.95~~ **$29.95** BW630

OK HOBBY WRAP-30 wire wrap & strip tool **$5.45**

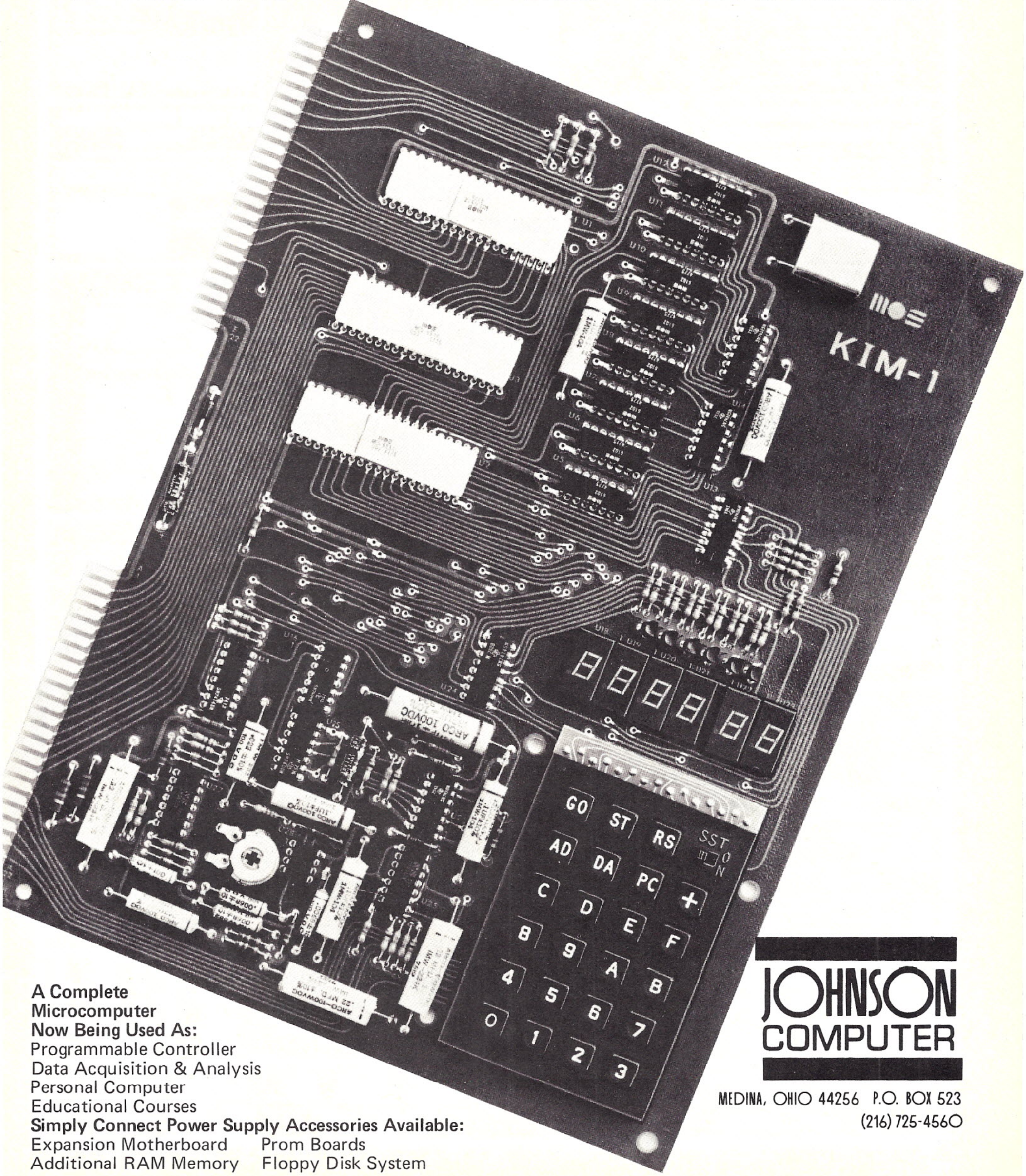
(213) 679-9001

VISA master charge THE INTERBANK CARD

All merchandise sold by California Industrial is premium grade.
Orders are shipped the same day received.
PLEASE INCLUDE $1.00 SHIPPING ON ORDERS UNDER $15.00
California residents add 6% sales tax • Money back guarantee.
Sorry, no COD's • Foreign orders add 10%
Credit cards accepted, $20 minimum

Reader inquiry card number C-50

KIM-1 245.00



A Complete Microcomputer Now Being Used As:

Programmable Controller
Data Acquisition & Analysis
Personal Computer
Educational Courses

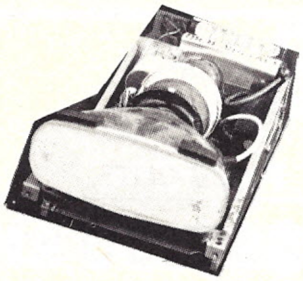
Simply Connect Power Supply Accessories Available:

Expansion Motherboard
Prom Boards
Additional RAM Memory
Floppy Disk System

JOHNSON COMPUTER

MEDINA, OHIO 44256 P.O. BOX 523
(216) 725-4560

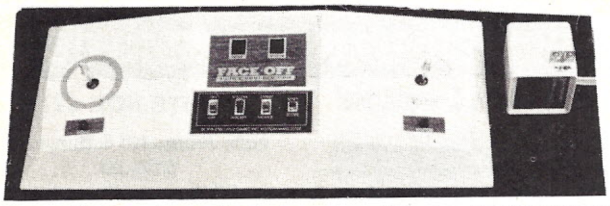
TYPESETTING MONITOR



We have acquired an unusual video monitor. It was originally designed for and used in the photo typesetting industry. As the typesetter composed his copy, the copy would appear on the screen. The tube is a flat type about 6½" x 2¼", since only one line of copy was scanned at a time. Most of the tubes have slight burn marks, but are still usable. We have all the circuitry, and will supply the circuitry with each order. We believe most of these units are in working order, since they were taken out of service when the equipment was upgraded. Looks like a good start to make into your own video monitor at giveaway prices. We have them with and without the CR tubes. Sold as is only.

STOCK No. 5603K Typesetter monitor with tube & circuit diagrams ...... $22.50 ea. 2/$40.00
STOCK No. 5604K Typesetter monitor without tube, with circuit diagrams . $15.00 ea. 2$25.00

TV HOCKEY and SOCCER GAME plus VIDEOCUBE

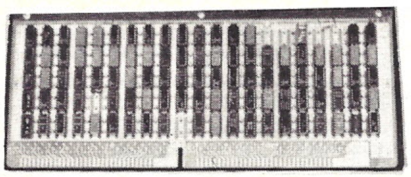


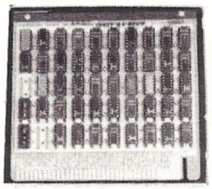
For some time we have been selling our VIDEOCUBE, the TV interface between computers, cameras, etc., and your TV set. We sold them in kit form, and to date we have sold over 3,000. We sold the complete VIDEOCUBE, when available, for $16.95.

We now have a limited number of Hockey-Soccer Video games, complete with VIDEOCUBE interface, at a new low price of $18.95; previously sold for $27.50 in kit form. This gives you two games, at two levels of skill, with LED readouts for scorekeeping. This game sold in Boston department stores for $79.50.

STOCK No. 5495K Complete VIDEO GAME & VIDEOCUBE $18.95 ea. 2/$35.00

WIRE WRAP BOARDS LOADED WITH 7400 SERIES ICs





Since last summer, we have been selling 2 wire wrap boards, Our Stock No. 6558K with approximately 100 sockets, and our Stock No. 6559K with approximately 45 sockets. These have been successful, based on your orders and reorders. We now have the same boards, but with the sockets still containing the original SN7400 series ICs that were used in the computer that these boards were designed for. We checked the value of these ICs, against the lowest price ICs in several Electronics magazine, and found that at the lowest possible surplus prices, the values of the ICs on the 100 socket board ran to over $40.00. A sample of some of the chips on the board we looked at are as f ollows: 74H87, 7486, 74107, 7451, 7400, 7404, 7495, 7493, 7492, 74193, 7489 and many others, to numerous to mention. Also on some boards, are a few linears, and phase locked loops. Not everyone needs every chip, but if you are working at all with TTL, this is a great opportunity to get an inventory of the most useful chips at a ridiculous price. We are selling the 100 socket board with about 100 chips, for $10.00 more than the board itself, and the 45 socket chip for $5.00 more than the board itself. We will also include with each board, 2 edge connectors with the 100 socket board, and 1 edge connector with the 45 socket board.

STOCK NO.6558K	75 to 100 socket board	$18.75 ea.	2/35.00
STOCK NO.6559K	45 to 50 socket board	$11.75	2/22.00
STOCK NO.6749K	75 to 100 socket wire wrap board with ICs and edge connectors	$28.75 ea.	2/55.00
STOCK NO.6750K	45 to 50 socket wire wrap board with ICs and edge connector	$16.75 ea.	2/32.00
STOCK NO.6603K	Edge connector for either board	$ 2.00 ea	3/5.00

MORE NEW TRANSFORMERS

TRANSFORMER 1. Primary 115 V, Secondary 1, 30V @ 2 A.Secondary 2, 16.5 V @ 1.2 A. Secondary 3, 16 V @ 3.5 A. Secondary 4, 9.5 V @ 3.5 A.

STOCK NO.6677R Wt. 10 lbs. $10.95 ea. 2/20.00

TRANSFORMER 2. Primary, 115 V. Secondary 1, 12 V. @ 5 A., Secondary 2, 24 V. @ 9 A., Seconday 3 14 V. @ 20 A. Secondary 4, 125 V. @ 1.5 A.

STOCK NO.6675K 16 lbs. 18.95 2/36.00

TRANSFORMER 3. Primary 115 V. Secondary 1, 12.6 V ct @ 1 A.Secondary 2, 33 V tapped at 13.1 V @ 1 Amp. Secondary 3, 140 V. @ 100 ma.

STOCK NO.6772K 2 lbs. 4 1/16"x 2 1/16"x 2 5/8". 2.95 ea. 2/5.00

MAIL ORDER

DELTA ELECTRONICS
P.O. BOX 2, 7 Oakland St.
Amesbury, Mass. 01913

Over the counter salesrooms

ATLANTA GA
DELTA ELECTRONIC HOBBIES
5151 Buford Hwy.
Doraville, Atlanta, Ga.

BOSTON MASS.
DELTA ELECTRONICS
WAREHOUSE OUTLET
590 Commonwealth Ave.
Boston, Mass.

D13

16K E-PROM CARD

New!

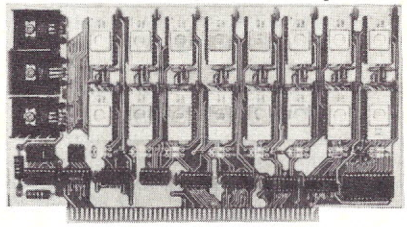
IMAGINE HAVING 16K OF SOFTWARE ON LINE AT ALL TIME!

S-100 (Imsai/Altair) Buss Compatible!

KIT FEATURES:

1. Double sided PC board with solder mask and silk screen and gold plated contact fingers.
2. Selectable wait states.
3. All address lines & data lines buffered!
4. All sockets included.
5. On card regulators.

KIT INCLUDES ALL PARTS AND SOCKETS (except 2708's). Add $25. for assembled and tested.



PRICE CUT!

$57.50 kit

WAS $69.95

SPECIAL OFFER:

Our 2708's (450NS) are $12.95 when purchased with above kit.

Fully Static!

8K LOW POWER RAM KIT-$149.00

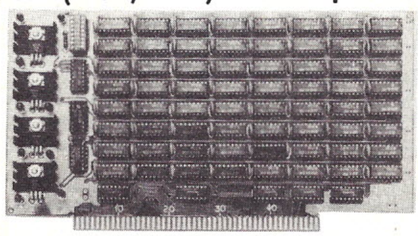
ADD $20 FOR 250NS

S-100 (Imsai/Altair) Buss Compatible!

KIT FEATURES:

1. Doubled sided PC Board with solder mask and silk screen layout. Gold plated contact fingers.
2. All sockets included.
3. Fully buffered on all address and data lines.
4. Phantom is jumper selectable to pin 67.
5. FOUR 7805 regulators are provided on card.

(450NS)



USES 21L02 RAM'S!

2 KITS FOR $279

Fully Assembled & Burned In
$179.00
Blank PC Board w/ Documentation
$29.95
Low Profile Socket Set 13.50
Support IC's (TTL & Regulators)
$9.75
Bypass CAP's (Disc & Tantalums)
$4.50

MOTOROLA QUAD OP - AMP
MC 3401. PIN FOR PIN SUB.
FOR POPULAR LM 3900.
3 FOR $1

ALARM CLOCK CHIP
N.S. MM5375AA. Six Digits.
With full Data. **New!**
$1.95 each

FULL WAVE BRIDGE
4 AMP. 200 PIV.
69¢ 10 FOR $5.75

NOT ASSOCIATED WITH DIGITAL RESEARCH OF CALIFORNIA, THE SUPPLIERS OF CPM SOFTWARE.

MOTOROLA 7805R VOLTAGE REGULATOR
Same as standard 7805 except 750 MA output.
TO—220. 5VDC output.
44c each or 10 for $3.95

450 NS! 2708 EPROMS
Now full speed! Prime new units from a major U.S. Mfg. 450 N.S. Access time. 1K x 8. Equiv. to 4-1702 A's in one package.
$15.75 ea. 4 FOR $50.00

COMPUTER MFG. MAKES $500,000.00 GOOF!

MOTOROLA 4K RAM CHIPS — 99¢ EACH **22 PIN DIP**

A major U.S. computer mfg. installed thousands of these 4K Motorola RAM's on the wrong boards and had to remove them. All parts were then tested, and met **FULL SPECS!** If you don't mind a little solder on the leads, then this is the best memory buy in the world. Arranged as 4096 x 1 Bits. 470 NS.
The Motorola 6605 is one of the easiest dynamic RAM's to use since it DOES NOT require multiplexed addresses as do most other 4K's such as the 4096 or 4027. A complete memory board design using the 6605 is outlined in the Motorola M6800 Applications Manual starting on page 4-70.

99¢ EACH (WITH DATA SHEET) **LOOK!**

MCM6605 **FOR $6.95 YOU GET AS MUCH STORAGE AS IN 32 - 2102's!** **8 FOR $6.95** 4096 BYTES OF RAM!

FULLY GUARANTEED! **SURPLUS BUY OF THE DECADE!!!**

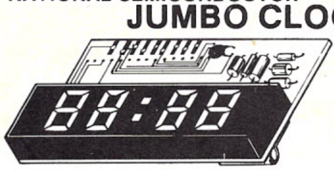
4K STATIC RAM'S
2114. The new industry standard. Arranged as 1K x4. Equivalent to 4-21 L02's in 1 package! 18 pin DIP. 2 chips give 1Kx8.
2/$24. 8/$85.

OPCOA LED READOUT
SLA-1. Common Anode. .33 inch character size. The original high efficiency LED display. **75c ea.**
4 FOR $2.50

Z-80 PROGRAMMING MANUAL
By Mostek, The major Z-80 second source. The most detailed explanation ever on the working of the Z-80 CPU CHIPS. At least one full page on each of the 158 Z-80 instructions. A MUST reference manual for any user of the Z-80. 300 pages. Just off the press. **$12.95**

NATIONAL SEMICONDUCTOR
JUMBO CLOCK MODULE

MA1008A. BRAND NEW!



$6.95

2 FOR $13

(AC XFMR $1.95)

FEATURES:
- FOUR JUMBO ½ INCH LED DISPLAYS
- 12 HR REAL TIME FORMAT
- 24 HR ALARM SIGNAL OUTPUT
- 50 OR 60 Hz OPERATION
- LED BRIGHTNESS CONTROL
- POWER FAILURE INDICATOR
- SLEEP & SNOOZE TIMERS
- DIRECT LED DRIVE (LOW RFI)
- COMES WITH FULL DATA

ASSEMBLED! NOT A KIT!

ZULU VERSION!
We have a limited number of the 24 HR Real time version of this module in stock.
#MA1008D — $9.95

PERFECT FOR USE WITH A TIMEBASE.

COMPARE AT UP TO TWICE OUR PRICE!

MANUFACTURER'S CLOSEOUT!

Tantalum Capacitors
1 MFD. .35V. By Kemet. Axial Lead.
Best Value! **10/$1.**

New! REAL TIME Computer Clock Chip
N.S. MM5313. Features BOTH 7 segment and BCD outputs. 28 Pin DIP. **$4.95 with Data**

MICRO-MINI TOGGLE SWITCH

SPDT. By RAYTHEON.
MADE IN USA! WITH HDWR.
99¢ EACH **6 FOR $5**

Digital Research Corporation
(OF TEXAS)
P. O. BOX 401247 • GARLAND, TEXAS 75040 • (214) 271-2461

TERMS: Orders under $15. add 75c. No COD's. We accept VISA, MasterCharge and American Express Cards. Money Back Guarantee on all items! Texas Residents add 5% Sales Tax. **WE PAY POSTAGE!**

FREE PROBES FREE PROBES FREE PROBES

NLS MS-215 DUAL TRACE MINISCOPE $395.00

NLS

1¢ SALE

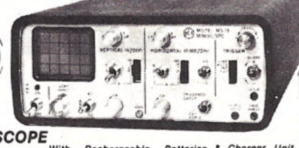
LM3A 3 dig 1% DC . . . . $125.
LM3.5A 3½ dig .5% DC . . . . $147.
LM40A 4 dig .1% DC. . . . . $190.
LM4A 4 dig .03% DC . . . . $227.

- Rechargeable batteries and charger included
- Measures DC Volts, AC Volts, Ohms and Current
- Automatic polarity, decimal and overload indication
- Rechargeable batteries and charger included
- Measures DC Volts, AC Volts, Ohms and Current
- Automatic polarity, decimal and overload indication
- No zero adjustment and no full-scale ohms adjust
- Battery operated — NiCad batteries; also AC line operation.
- Large LED display for easy reading without interpolation
- Size: 1.9"H x 2.7"W x 4"D

Parts & labor guaranteed 1 year
Tilt stand option . . . . $ 3.50
- Leather case $16.00

Purchase any of the LM series Meters and buy the LEATHER CASE for 1¢

LAST CHANCE: NLS PRICES GOING UP



NLS

$289

MS-15 MINISCOPE *With Rechargeable Batteries & Charger Unit*

- 15 megahertz bandwidth.
- External and internal trigger.
- Time base — .1 microsec to 0.5 sec/div - 21 settings . ± 3%.
- Battery or line operation.
- Automatic & line sync modes.
- Power consumption ... 15 watts.
- Vertical gain — ... 50 V/div - 12 settings ± 3%.
- Viewing area 1... x 1.35".
- Case size 2.... H x 6.4"W x 7.5"D, 3 pounds.
- Parts & labor guaranteed 1 year
- 10... 10 meg probe $24.50
- Leather carrying case $30.00
- MS-215 Dual Trace Version of MS-15 $395.00

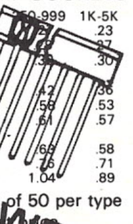
PROBE 1¢

PROBE 1¢ with the purchase of SCOPE and the MENTION of this MAGAZINE

NLS

3 LEVEL GOLD WIRE WRAP SOCKETS

	1-24	25-49	50-99	100-249	250-999	1K-5K
8 pin*	.41	.38	.35	.31		.23
14 pin*	.39	.38	.36	.32		.27
16 pin*	.4	.42	.39	.35	.3	.30
18 pin	.63	.58	.54	.47	.42	.36
20 pin	.80	.75	.70	.68	.58	.53
22 pin	.90	.85	.80	.70	.61	.57
24 pin	.90	.84	.78	.68	.63	.58
28 pin	1.10	1.00	.90	.84	.76	.71
40 pin	1.50	1.40	1.3	1.20	1.04	.89



3 LEVEL GOLD WIRE WRAP SOCKETS

Sockets purchased in multiples of 50 per type may be combined for best price.

All sockets are GOLD 3 level closed entry *End and side stacable. 2 level, Solder Tail, Low Profile, Tin Sockets and Dip Plugs available. CALL FOR QUOTATION

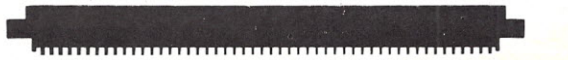
SALE S-100 BUS EDGE CONNECTORS SALE

S100-WWG 50/100 Cont. .125 ctrs. 3 LEVEL WIRE WRAP .025" sq. posts on .250 spaced rows. GOLD plated.

1-4	5-9	10-24
$4.00	$3.75	$3.50

S100-WWN 50/100 Cont. .125 ctrs. 3 LEVEL WIRE WRAP .025" sq. posts on .250 spaced rows. NASGLO tin-nickel plated.

1-4	5-9	10-24
$3.50	$3.25	$3.00



S100-STG 50/100 Cont. .125 ctrs. DIP SOLDER TAIL on .250 spaced rows for VECTOR and IMASI motherboards GOLD plated.

1-4	5-9	10-24
$4.00	$3.75	$3.50

S100-STN 50/100 Cont. .125 ctrs. DIP SOLDER TAIL on .250 spaced rows for VECTOR and IMASI motherboards. NASGLO tin-nickel plated.

1-4	5-9	10-24
$3.50	$3.25	$3.00

RG81G 50/100 Cont. .125 ctrs. DIP SOLDER TAIL on .140 spaced rows for ALTAIR motherboards. GOLD plated. $5.00

R681-3 50/100 Cont. .125 ctrs. PIERCED SOLDER EYELET tails. GOLD $7.35

Other Popular Edge Connectors

R644-G 22/44 Cont. .156 ctrs. PIERCED SOLDER EYELES tails. GOLD plated.

1-4	5-9	10-24
$3.00	$2.75	$2.50

R644-3 22/44 Cont. .156 ctrs WIRE WRAP tails. GOLD $4.71

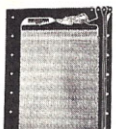
ATTN: OEM'S and Dealers, many other connectors available call or quotation.

NLS MS-215 DUAL TRACE MINISCOPE $395.00

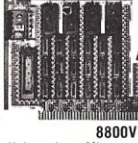
- Kit includes 12 tantalum capacitors for +5, +12, —12 buses and insulated mounting spacers.
- Wiring side shown. Component side bare epoxy glass with white markings for component locations.
- G10 epoxy glass board with 2 ounce copper, solder plated and .038 diameter holes for leads.
- Solder mask with solder windows on etched circuits to avoid accidental short circuits.
- Mounts 11 receptacles with 100 contacts (2 rows) on .125 centers with .250 row spacing. Vector part number R681-2; or mounts 10 receptacles plus interconnections to smaller mother board for expansion.
- Includes etched circuits and instructions for option of active, pull-up, or floating terminations.
- Large buses: +5V and GND (10 AMPS), ± 12V or 16V (7 AMPS). Current ratings are per MIL-STD-275 with 10°C rise.
- Fits in Vector-pak enclosures.
- Fits in IMSAI 8080 microcomputer as expander board.

8803 MOTHER BOARD FOR S100 BUS MICRO-COMPUTERS

Vector



Price: $29.50



Vector **Plugboards**

8800V

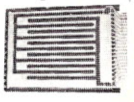
Universal Microcomputer/processor plugboard, use with S-100 bus. Complete with heat sink & hardware. 5.3" x 10" x 1/16"

1-4	5-9	10-24
$19.95	$17.95	$15.96

8801-1

Same as 8800V except plain; less power buses & heat sink.

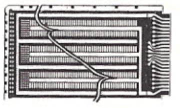
1-4	5-9	10-24
$14.95	$13.46	$11.96



3682 9.6" x 4.5" **$10.97**

3682-2 6.5" x 4.5" **$9.81**

Hi-Density Dual-In-Line Plugboard for Wire Wrap with Power & Grd. Bus Epoxy Glass 1/16" 44 pin con. spaced .156



3677 9.6" x 4.5" **$10.90**

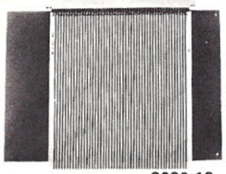
3677-2 6.5" x 4.5" **$9.74**

Gen. Purpose D.I.P. Boards with Bus Pattern for Solder or Wire Wrap. Epoxy Glass 1/16" 44 pin con. spaced .156

3662 6.5" x 4.5" **$7.65**

3662-2 9.6" x 4.5" **$11.45**

P pattern plugboards for IC's Epoxy Glass 1/16" 44 pin con. spaced .156



3690-12 CARD EXTENDER

Card Extender has 100 contacts-50 per side on .125 centers-Attached connector-is compatible with S-100 Bus Systems. $25.00

3690 6.5" 22/44 pin .158 ctrs. Extenders $12.00

1/16 Vector BOARD .042 dia holes on 0.1 spacing for IC's

PART NO.	SIZE	PRICE 1 - 9	10 - 19
Phenolic			
64P44XXXP	4.5x6.5"	$1.49	1.34
169P44XXXP	4.5x17"	$3.51	3.16
Epoxy Glass			
64P44	4.5x6.5"	$1.70	1.53
84P44	4.5x8.5"	$2.10	1.89
169P44	4.5x17"	$4.30	3.87
169P84	8.5x17"	$7.65	6.89

SLIT-N-WRAP

Wraps insulated wire on .025" square posts
FOUR TIMES FASTER than regular manual wrap-post tools

P180 with two 100' spools of 28 ga. wire **$24.50**

P160-4T includes charger, wire **$75.00**

Vector

NO PRE-STRIPPING*
NO PRE-CUTTING*
SPOOL-FED WIRE*

*The spooled wire passes through the tool past a slitting edge next to the wrap post. A narrow longitudinal cut is made in the insulation where it presses the square post corner, the bared copper is indented by the sharp edge. (7 turns = 28 contacts)

Insulation is slit where wrapped but not between terminal posts when led straight out of the tool

"Daisy-chain" runs
Reliable and easy to use
Manual or power operation

SLIT-N-WRAP WIRE NO. 28 GAGE INSULATED WIRE, 100' SPOOLS $2.75 Package

W28-2-A-Pkg. 3. Green W28-2-C-Pkg. 3. Clear
W28-2-B-Pkg. 3. Red W28-2-D-Pkg. 3. Blue

2708

8K 450 ns

EPROM

FACTORY PRIME

$10.00 EA.

25 + Call For Price

WOW 14 & 16 PIN GOLD 3 LEVEL WIRE WRAP SOCKETS

14 - G3 100 for **$30.00**

16-G3 100 for **$30.00**

50 of each for **$32.00**

Sockets are End & Side stackable, closed entry

LIQUID CRYSTAL DIGITAL CLOCK-CALENDAR

- For Auto, Home, Office
- Small in size (2x2½x½)
- Push button for seconds release for date.
- Clocks mount anywhere with either 3M double-sided tape or VELCRO, included.
- 2 MODELS AVAILABLE:
  LCD-101, portable model runs on self-contained batteries for better than a year.
  LCD-102, runs on 12 Volt system and is back-lighted.
- LCD-101 or LCD-102 your choice. . . . . **$34.95 ea.**
- Clear desk stand for **$2.00**

LEDU MG 10A List $72.

Perfectly balanced, fluorescent lighting with precision magnifier lens. For prof'l, techn'l & hobbist. Has die cast protective shade, inst. start 3 diopter lens. 42" reach.

$44.95

with T-9 fluorescent lamp (required) $49.95
colors: Gray or Black
Chocolate Brown
2 choices please

$44.95 SPECIAL

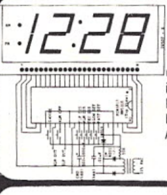
SC-5 With Rechargeable Batteries & Charger Unit $89

FM-7 With Rechargeable Batteries & Charger Unit $195

NLS

Features Include: • By using the new NLS SC-5 Prescaler, the range of the FM-7 Frequency Meter, which is 10 Hz to 60 MHz, may be extended to 512 MHz (the upper VHF & UHF frequency bands). • The FM-7 utilizes an LED readout, providing 7-digit resolution. • The FM-7 can be calibrated to an accuracy of 0.00001%. • The SC-5 is accurate to one part per million. • Each unit has 30 millivolts sensitivity, is battery powered and has a charger unit included. • Dimensions of each are 1.9" H x 2.7" W x 3.9" D. • The units may be obtained separately or as a "Frequency Duo." • Parts & Labor guaranteed 1 year.
Tilt stand option. $ 3.50
Leather case $16.00

MICRO-KLIP
for .042 dia. holes
(all boards on this page)
T42-1 pkg. 100 . . **$ 1.50**
T42-1 pkg. 1000 . . . . . . **$11.00**
P-149 hand installing tool . . . . . . **$ 2.03**

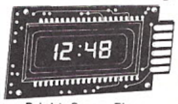


.8" LED ALARM CLOCK

12 hr. LED Alarm Clock uses 3½ digit .8" LED Display with AM/PM indicators and colons. Direct drive, PIN to PIN interface with S1998A I.C. Just add switches, AC Supply, Alarm. Display and I.C. only.

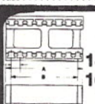
$7.95 or 2/$15.00

Price Breakthrough! **$17.50**



MA1003 CAR CLOCK

Bright Green Fluorescent Display Crystal Time Base Assembled, just add switches and 12 VDC.



SPECIAL

14CS2 100 for $14.00
16CS2 100 for $16.00
14 pin CS2 10 for $2.00
16 pin CS2 8 for $2.00

These low cost DIP sockets will accept both standard width plugs and chips.
For use with chips, the sockets offer a low profile height of only .125" above the board. These sockets are end stackable.

Vector

WRAP POST
for .042 dia. holes
(all boards on this page)
T-44 pkg. 100 . . **$ 2.28**
T-44 pkg. 1000. . . . . . **$14.00**
A-13 hand installing tool . . . . . . **$ 2.80**

PRIORITY ONE ELECTRONICS ©

P21

VISA

4911K West Rosecrans, Hawthorne, CA 90250

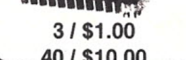
master charge

Terms: VISA, MC, BAC, check, Money Order, C.O.D., U.S. Funds Only. CA residents add 6% sales tax. Minimum order $10.00. Orders less than $75.00 include 10% shipping and handling; excess refunded. Just in case . . . please include your phone no. **"Sorry, no over the counter sales"**

Prices subject to change without notice.

Send for our latest brochure. **phone orders welcome (213) 973-4876** OEM and Institutional inquiries invited.

24 PIN DIP PLUGS WITH COVERS



3 / $1.00
40 / $10.00

FREE PROBES FREE PROBES FREE PROBES

POLY PAKS

COMPUTER PERIPHERAL & PARTS WORKSHOP

MAIL ORDER

POLY PAKS INTRODUCES THE FIRST MAGAZINE RETAIL STORE FOR THE ELECTRONIC MAIL-ORDER HOBBYIST!

POLY PAKS SPECIAL

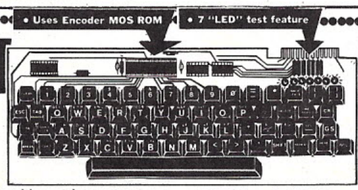
BUY $15 WORTH Take 10% DISCOUNT

BUY $100 WORTH Take 25% DISCOUNT

FROM THIS AD ONLY

$65. NEW KEYBOARD & ENCODER KIT

• Uses Encoder MOS ROM • 7 "LED" test feature



Cat. No. 8K5001	Kit	$65.00
Cat. No. 8K5002	Wired	$69.95

Outputs standard 7 bit ASCII; interfaces with most data systems. Keyboard pre-assembled onto PC board, 2 key rollover. Electronic shift lock and carriage return. 4 modes: Normal, control, shift, shift/control. Additional functions can be assigned by user. +5, -12VDC, 200 ma. Negative or positive logic, jumper selectable. Exclusive test feature... 7 LED's display the ASCII code. Complete kit, nothing else to buy! Size: 13" x 5 1/2 x 1 1/4". 3 lbs.

- Uses MOS Encoder ROM!
- 7 LED Test Feature!
- 64 key keyboard!
- Encodes128 ASCII Characters
- Interfaces with ALTAIR, IMSAI, and more!

HANDY COMPUTER MULTITESTER

$8.88

- 1000 ohms per volt

1% precision, movements diode protected against burnout. Measures DC volts 0-15-150-1000; AC volts 0-15-150-1000; DC current 0-150ma; resistance X1000. Sensitivity 1000 ohms/volt AC-DC. Uses penlite cell, not included. Size 2 3/8 x 3 1/2 x 1 3/8". Wt. 5 ozs. Cat. No. 8K3921

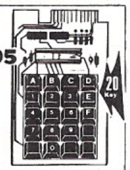
MICROPROCESSORS! MEMORIES! SUPPORT!

Order by Cat. No. 8K3459 and type

Type	Description	Sale
Z-80A	8 Bit CPU	$34.95
MC6800L	8Bit CPU	19.95
8080A	8 Bit CPU, 2 usec	12.95
8008	CPU	7.95
1101	256 x 1 Static RAM	.69
1103	1K Dynamic RAM	1.29
2102-L1	1K x 1 Lo-power RAM	1.69
2111	256 x 4 Static RAM	5.95
2112	256 x 4 Static RAM	2.49
1702A	256 x 8 EPROM	3.95
2708	8K EPROM	12.95
MK4200P11	4K Dyn RAM, 350 nsec	3.95
MK4096	4K Dyn RAM	3.95
MK4116	16K Dyn RAM	29.95
MM5202	2K PROM	2.95
MM5203	2K EPROM	3.95
MM5260	1K Dyn RAM	.99
MM5262	2K x 1 Dyn RAM	.99
8212	8 Bit I/O Port	3.95
8216	Bi-Direct Bus Driver	3.95
8224	Clock Gen	4.95
8228	System Cont	8.95
8251	Communication Int	11.50
8255	Periph Interface	11.95

HEXADECIMAL MICROPROCESSOR AND CONTROL KEYBOARD KIT! 34.95

Wt. 2 lbs.



Address microprocessors, control computer operated equipment. 2 key rollover. Has 20 keys, 16 encoded, 4 external to be assigned by user. Output 4 bit binary. Also, an EXCLUSIVE FEATURE... 4 leds display the binary output. TTL/CMOS compatible, requires +5, 12VDC. Complete kit! Nothing else to buy! With instructions.

- 4 bit binary/hexadecimal output plus strobe!
- New! Improved design! Saves you time!
- Keyboard pre-assembled onto PC board!

Cat. No. 8K5009	Hexadecimal Kit	$34.95
Cat. No. 8K5010	Hexadecimal Wired	$39.95

FULL WAVE BRIDGE RECTIFIERS!

PIV	2 AMP (#8K1346)	6 AMP (#8K2456)	10 AMP (#8K2447)	25 AMP (#8K2273)
50	$ .59	$ .88	$1.05	$1.20
100	.65	.99	1.15	1.25
200	.69	1.19	1.29	1.95
400	.89	1.40	1.79	2.95
600	.99	1.69	1.95	3.95
800	1.19	1.95	2.25	4.95
1000	1.25	2.25	2.50	5.50

Orer by Cat. No. Amperage and Voltage

MIX-N-MATCH DUAL POWER SUPPLIES As low as $16.88

- 1 Amp each voltage!
- 2 Amps total!

Cat. No.			Sale
8K3995	+5	-5	$16.88
8K3996	+5	-12	17.88
8K3999	+12	-12	18.88
8K4003	+15	-15	19.88

A new concept in power supply design. Choose from any one of four popular positive and negative combinations. Uses LM340 & LM320 series regulators and large heat sinks. Fits easily into gadget boxes or cabinet of your own design. Easy to assemble with step-bystep instructions. (Includes PC board, transformers, line cord, capacitors, rectifiers, heat sinks and h'ware). Complete kit, nothing else to buy. Wt. 10 lbs.

TTL'S

Type	Sale	Type	Sale
SN7400	$.14	SN7483	.55
SN7401	.14	SN7486	.29
SN7403	.14	SN7488	.79
SN7404	.14	SN7490	.45
SN7405	.14	SN7491	.59
SN7406	.14	SN7492	.39
SN7410	.14	SN7493	.39
SN7411	.19	SN7495	.49
SN7413	.49	SN7496	.29
SN7416	.25	SN74100	.89
SN7417	.24	SN74107	.29
SN7420	.19	SN74109	.65
SN7421	.25	SN74113	.39
SN7423	.25	SN74121	.31
SN7426	.19	SN74123	.49
SN7430	.14	SN74126	.49
SN7432	.24	SN74132	.71
SN7437	.24	SN74145	.69
SN7438	.24	SN74151	.69
SN7440	.19	SN74153	.69
SN7443	.49	SN74154	.99
SN7445	.69	SN74155	.79
SN7446	.65	SN74157	.59
SN7447	.65	SN74161	.79
SN7448	.65	SN74163	.79
SN7450	.14	SN74164	.79
SN7451	.14	SN74165	.99
SN7454	.19	SN74166	1.19
SN7460	.19	SN74173	.99
SN7462	.29	SN74174	.89
SN7470	.14	SN74175	.69
SN7472	.19	SN74177	.79
SN7473	.35	SN74179	1.99
SN7474	.35	SN74180	.49
SN7475	.39	SN74190	1.19
SN7476	.39	SN74192	.69
SN7480	.19	SN74193	.69
		SN74194	.79
		SN74197	.75

Low Power ICs

Order By Cat. No. 8K3667 & Type No.

Type	Sale	Type	Sale
74LS00	$ .29	74LS132	.99
74LS02	.29	74LS138	.99
74LS04	.29	74LS139	.99
74LS08	.29	74LS151	.89
74LS10	.29	74LS153	.99
74LS11	.36	74LS155	1.25
74LS13	.59	74LS160	.99
74LS14	.99	74LS161	.99
74LS20	.29	74LS162	.99
74LS21	.36	74LS163	.99
74LS22	.36	74LS168	1.25
74LS27	.39	74LS169	1.25
74LS30	.29	74LS173	1.25
74LS32	.36	74LS174	.99
74LS37	.39	74LS190	1.75
74LS38	.39	74LS191	1.75
74LS42	.79	74LS192	1.50
74LS47	.99	74LS193	1.25
74LS74	.49	74LS195	1.25
74LS90	.89	74LS196	1.25
74LS92	.89	74LS197	1.25
74LS93	.89	74LS257	.99
74LS109	.49	74LS258	.99
74LS112	.49	74LS266	.39
74LS113	.49	74LS366	.69
74LS114	.49	74LS368	.69
		74LS390	1.95

C-MOS

Order By Cat. No. 8K2320 & Type No.

Type	Sale	Type	Sale
CD4000	$ .25	CD4023	.25
CD4001	.25	CD4024	.99
CD4002	.25	CD4025	.25
CD4006	1.20	CD4027	.49
CD4007	.25	CD4028	1.10
CD4008	1.20	CD4029	1.20
CD4009	.59	CD4030	.45
CD4010	.59	CD4032	.75
CD4011	.25	CD4035	1.40
CD4012	.25	CD4040	1.40
CD4013	.45	CD4041	.99
CD4014	.99	CD4042	.89
CD4015	1.20	CD4044	.99
CD4016	.45	CD4046	1.50
CD4017	1.20	CD4047	2.50
CD4018	1.20	CD4049	.45
CD4019	.35	CD4053	.89
CD4020	1.25	CD4066	.90
CD4021	1.25	CD4071	.25
CD4022	1.25	CD4072	.39
		CD4077	.69

COMPUTER GRADE CAPACITORS

Order by Cat. No. 8K5112 and value!

Mfd	WVCD	Sale
8,000	50	$2.75
13,000	40	2.25
14,000	50	2.75
15,000	12	1.50
22,000	75	3.95
24,000	30	2.50
44,000	30	3.95

DISCRETE LED'S 6 for $1.19

Order in multiples of 6 of each type!

Size	Cat. No.	Description	Similar to
	8K2135	Jumbo Red	MV5053
	8K1944	Jumbo Yellow	XC556Y
	8K2138	Jumbo Green	XC556G
JUMBO .34" h	8K2785	Medium Red	XC22
MEDIUM .24"	8K2788	Medium Green	XC22G
	8K2137	Micro Red	XC209
MICRO .21"	8K1948	Micro Yellow	XC209Y
(TO-18)	8K2140	Micro Green	XC290G

SILICON POWER STUD RECTIFIERS!

Order by Cat. No. Amperage and voltage.

PIV	12 AMP (#8K727)	50 AMP (#8K727)	250 AMP (#8K685)
50	$ .29	$ .75	$4.95
100	.36	1.10	6.50
200	.45	1.35	7.50
400	.61	1.75	8.50
600	.79	2.25	10.50
800	.90	2.75	11.50

"RED" LED READOUTS!

as low as 20¢ each

Cat. No.	Size	Description	Sale
8K1890	.127"	MAN-3**	5 for $1.00
8K1503	.19	MAN-4*	2 for $1.19
8K1273	.27	MAN-1*	$1.00
8K3093	.33	MAN-72 equal*	$1.19
8K3161	.33	MAN-74 equal**	$1.19
8K3512	.35	FND359**	$1.00
8K2949	.5	FND500*	$1.50
8K2950	.5	FND507*	$1.50
8K3483	.5	727-Dual**	$2.50
8K2485	.5	727-Dual*	$2.50
8K2256	.6	747*	$1.95

*Common Anode **Common Cathode

POLY PAKS "CHIPS" AWAY IC AND CRYSTAL PRICES!

Order by Cat. No. 8K4048 and Type No.

Type	Description	Sale
ICM7205	Stopwatch	$14.95
AY3-8500-1	6 TV Games	9.95
AY5-9100	Telephone Dialer	12.95
AY5-9200	Repertory Dialer	9.95
MM5330	4 1/2 Digit DVM	4.95
8038	Volt Control Osc	4.95
KR2376-139	BCD Encoder ROM	6.95
KR2376-ST	ASCII Encoder Rom	14.95
11C90DC	650MHz Prescaler	14.95
95H90DC	350MHz Prescaler	8.88
MC14410	Touch Ton Chip	9.95
MK2002P	Char Gen. (sim 2513)	3.95
7207A/7208	Freq. Counter pair	24.95
ICM7107	3 1/2 Digit DVM/DMM	12.50
MM5369	60Hz Timebase IC	1.95
3.579MHz	Timebase Crystal	1.95
1.00MHz	Touchtone Crystal	5.95
3.2768MHz	Stopwatch Crystal	4.95
ULN2300	Op Amp With Built-In SCR	1.19
RC4195	Dual 5 Volt Tracking VR	2.95
TVR2000	High Power 723 Volt Reg	.59

DIP SWITCHES

Cat. No.	Switches	Sale
8K 3668	2	$.77
8K 3669	3	.88
8K 3021	4	.99
8K 3671	6	1.29

EECO 10-POSITION BCD THUMBWHEEL SWITCH

1-2-4-8 BCD encoding! Positions labeled 0-7, F, L. (F coded for 8, L coded for 9). White numerals on a black background. Eeco 800 series. 1 1/4 x 1 1/4 x 1/2". Cat. No. 8K3846 3 for $4. $1.49

LINEARS

Type	Sale	Type	Sale
LM301V,H	$.25	LM380N	1.19
LM307V,N,H	.25	LM381N	1.19
LM308V	.19	LM555V	.44
LM309K	.99	LM558V	.19
LM311V	.19	LM565N,H	.65
LM320T-6	1.19	LM567V	1.40
LM340T-5	1.19	LM703H	.39
LM340T-12	1.19	LM709H	.19
LM340T-15	1.19	LM710N	.29
LM340K-5	1.19	LM741V,H	.29
LM340K-12	1.19	LM747H	.59
LM340K-15	1.19	LM1458V	.29
LM340K-18	1.19	LM1800N	.19
LM340K-24	1.19	LM3900N	.19
LM322N	1.19	LM3909V	.89
LM324N	.99	75491	.69
LM339N	1.09	75492	.75
LM377N	1.50	PA263	1.50

COMPUTER GRAPHICS "JOYSTICK" 2 for $8. $4.50

- Four 100K pots!
- Chrome handle and knob!

Wt. 10 oz. Cat. No. 8K3808A

IC SOCKETS

as low as 17¢ ea.

- Low profile, solder tail.

Cat. No.	Description	Sale
8K2123	8 pin minidip	$ .17
8K1308	14 pin dip	.19
8K1309	16 pin dip	.22
8K3378	18 pin dip	.25
8K5265	14 pin Wire Wrap*	.29
8K5266	16 pin Wire Wrap*	.35
8K3886	24 pin MSI	.59
8K3887	28 pin MSI	.59
8K3802	40 pin LSI	.99

*Gold Plated

GIANT SALE! MICRO-MINI TOGGLE SWITCHES

- 3A, 125VAC contacts or better
- Complete with mounting hardware!

Cat. No.	Contacts	Sale
8K4036	SPDT	$1.29
8K5085	SPDT*	1.39

*Center off

RIBBON CABLE AT THIN PRICES

- Ultra-flat! • 28 AWG!
- Single color! Indexed!

Cond.		Sale
20	8 ft.	$1.98
26	6 ft.	$1.98
34	5 ft.	$1.98
40	4 ft.	$1.98

Order by Cat. No. 8K3939 and conductors

SPECTRA-TWIST RIBBON CABLE

- Twisted pairs of brightly colored cable! 24AWG

Cat. No. 8K3680 48 cond. 2 ft. $1.98

WOW! MOTHERBOARD EDGE CONNECTOR

106 pins (53 each side). Use with IMSAI & ALTAIR! 0.125" pin ctrs, goldplated 5/8" wire wrap leads. Open ends, fit wide PC boards. 8 ozs. Cat. No. 8K3987

$2.50

$9.95 IMC "PEWEE" BOXER FAN

- Smallest Box Fan Made!
- Only 3 1/8" SQ, 1 5/8" Deep!

Cat. No. 8K5249

Quiet! Dependable! Powerful! Delivers 36 cfm Scarcely Audible Noise Level! Sleeve bearings rated for 10 years life! Perfect for cooling, ventillation, heating: for computers, power supplies, hi-fi eqpt, light projectors & more. 7-Blade flame-retardant polycarbonate fan, housed in die cast aluminum block. Wt. 1 lb.

For Hi-Fi and Computers. 115 VAC!

10 AMP POWER TAB SCR'S, TRIACS, QUADRACS!

Order by Cat. No. and Voltage

	Cat. No.	PIV	SALE
SCR'S	8K1730	50	$.55
TRIACS	8K1448	100	.69
QUADRACS	8K1590	200	.88

TRANSFORMER SALE

LOOK!

• 110VAC Primaries

Cat. No.	Output V.	Amps		Sale Each
8K 3399	6.3V	500ma	Metal encased	$1.98
8K 3814	6.3V	1A	Open frame	$2.49
8K 3412	12V	300ma	Open frame	$1.95

1N4000 Epoxy Rectifiers

- Miniature! • 1.5 Amp!

Order by Cat. No. and Type No.

Cat. No.	Type No.	Volts	Sale	
8K2377	1N4001	50	10 for	$.75
8K2378	1N4002	100	10 for	.85
8K2379	1N4003	200	10 for	.95
8K2380	1N4004	400	10 for	1.19
8K2381	1N4005	600	10 for	1.39
8K2382	1N4006	800	10 for	1.49
8K2383	1N4007	1000	10 for	1.59

WRITE FOR POLY PAKS 1978 CATALOG

FREE!

FEATURING:

- Computer Components!
- Over 400 Assortments!
- Digital Clocks, DPM's, and Stopwatches!
- Hi-Fi Stereos-Speakers!
- Solar Energy! Semi's!
- Test Equipment!
- Fiber Optics! Plus More n' More!

Terms: Add postage Rated: net 30
Phone: Wakefield, Mass. (617) 245-3829
Retail: 16-18 Del Carmine St., Wakefield.

MINIMUM ORDER — $6.00

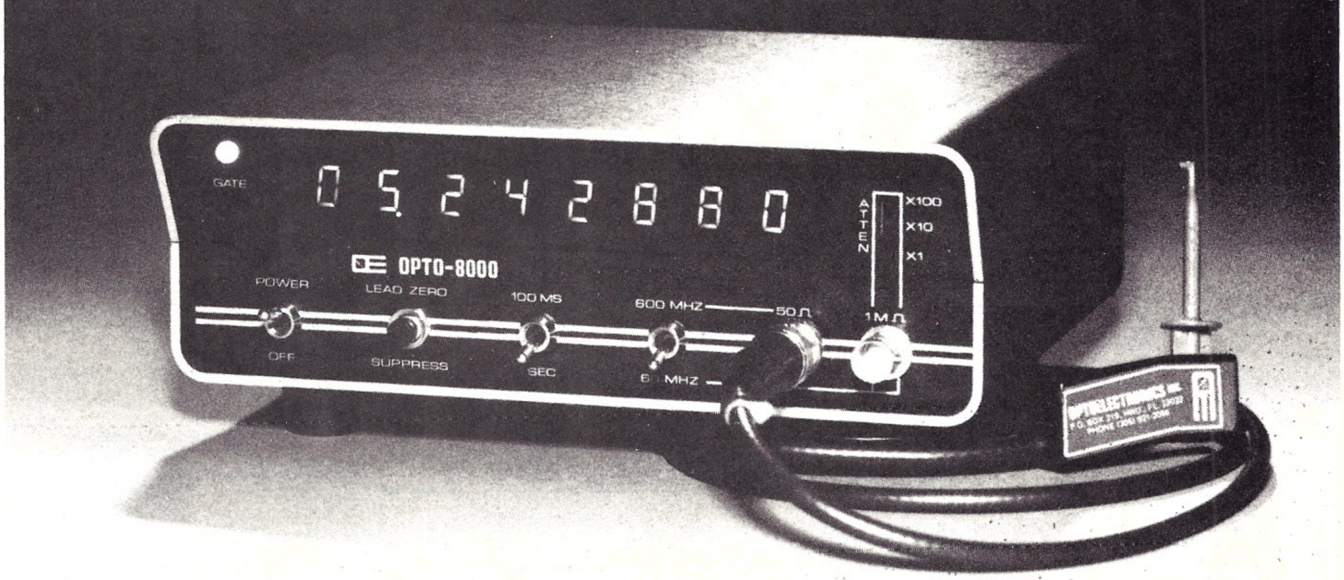
POLY PAKS®

C.O.D.'s MAY BE PHONED

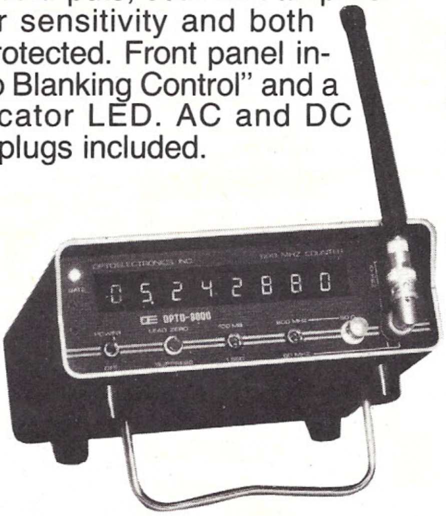
P.O. BOX 942-K8
LYNNFIELD, MA.
ZIP 01940

600 MHZ. FREQUENCY COUNTER
±0.1 PPM TCXO

OPTO-8000.1



This new instrument has taken a giant step in front of the multitude of counters now available. The Opto-8000.1 boasts a combination of features and specifications not found in units costing several times its price. Accuracy of ±0.1 PPM or better — *Guaranteed* — with a factory-adjusted, sealed TCXO (Temperature Compensated Xtal Oscillator). **Even kits require no adjustment for guaranteed accuracy!** Built-in, selectable-step attenuator, rugged and attractive, black anodized aluminum case (.090″ thick aluminum) with tilt bail. 50 Ohm and 1 Megohm inputs, both with amplifier circuits for super sensitivity and both diode/overload protected. Front panel includes "Lead Zero Blanking Control" and a gate period indicator LED. AC and DC power cords with plugs included.



SPECIFICATIONS:
Time Base—TCXO ±0.1 PPM GUARANTEED!
Frequency Range—10 Hz to 600 MHz
Resolution—1 Hz to 60 MHz; 10 Hz to 600 MHz
Decimal Point—Automatic
All IC's socketed (kits and factory-wired)
Display—8 digit LED
Gate Times—1 second and 1/10 second
Selectable Input Attenuation—X1, X10, X100
Input Connectors Type —BNC
Approximate Size—3″h x 7½″w x 6½″d
Approximate Weight—2½ pounds
Cabinet—black anodized aluminum (.090″ thickness)
Input Power—9-15 VDC, 115 VAC 50/60 Hz
or internal batteries
OPTO-8000.1 Factory Wired **$299.95**
OPTO-8000.1K Kit **$249.95**

ACCESSORIES:
Battery-Pack Option—Internal Ni-Cad Batteries and charging unit **$19.95**
Probes: P-100—DC Probe, may also be used with scope **$13.95**
P-101—LO-Pass Probe, very useful at audio frequencies **$16.95**
P-102—High Impedence Probe, ideal general purpose usage **$16.95**

VHF RF Pick-Up Antenna-Rubber Duck w/BNC #Duck-4H **$12.50**
Right Angle BNC adapter #RA-BNC **$ 2.95**

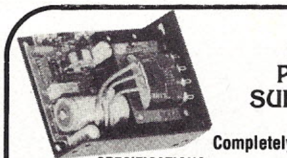
FC-50 — Opto-8000 Conversion Kits:
Owners of FC-50 counters with #PSL-650 Prescaler can use this kit to convert their units to the Opto-8000 style case, including most of the features.
FC-50 — Opto-8000 **Kit $59.95**
*FC-50 — Opto-8000F **Factory Update $99.95**
FC-50 — Opto-8000.1 (w/TCXO) **Kit $109.95**
*FC-50 — Opto-8000.1F **Factory Update $149.95**
*Units returned for factory update must be completely assembled and operational

OPTOELECTRONICS, INC.
5821 NE 14 Avenue
Ft. Lauderdale, FL 33334
Phones: (305) 771-2050 771-2051
Phone orders accepted 6 days, until 7 p.m.

03

master charge
BANKAMERICARD

TERMS: Orders to U.S. and Canada, add 5% to maximum of $10.00 per order for shipping, handling and insurance. To all other countries, add 10% of total order. Florida residents add 4% state tax. C.O.D. fee: $1.00. Personal checks must clear before merchandise is shipped.



ELPAC POWER SUPPLIES

Completely Assembled

SPECIFICATIONS:

105-125/210-250 Vac, 47-440 Hz Input:

Line Regulation	±0.1%
Load Regulation	±0.1%no-load to rated-load
Output Ripple and Noise	±0.1%p-p,dc to 10 MHz
Input/Output Isolation	100 megohm dc, 900 Vac
Short Circuit Current	35% rated current

PART NO.	RATINGS WATTS	VOLTS	AMPS	PRICE
SOLV15-5*	15	5	3	$36.95
SOLV15-12*	15	12	1.5	36.95
SOLV30-5	30	5	6	59.95
SOLV30-12	30	12	3	59.95
OVP1	over voltage protection for SOLV30-5,-12			9.95

*SOLV15-5, 12 includes OVP installed

NEW! BULB-ENERGY SAVER

Bulb lasts 3 or more times longer. Fits Standard Socket 6 watts to 200 watts

BULB-ENERGY SAVERS used for years by major industrial users — now available for home or office use. Bulb Savers can cut electrical bills by as much as 3%.
BULB-SAVERS lengthens light life by:

1. Acting as an electrical "shock absorber", turns the bulb on slowly, eliminating the "thermal shock". Bulb life increases 300 percent.
2. Banishes Current "Surges". Cushions line voltage surges when other loads cut power line.
3. Reduces Energy Consumption

	1-9	10+
BES-1	1.39 ea.	1.20

CRYSTALS

THESE FREQUENCIES ONLY

PART NO.	FREQUENCY	CASE	PRICE
CY1A	1.000MHz	HC33	5.95
CY1.84	1.8432MHz	HC33	5.95
CY2A	2.000MHz	HC33	5.95
CY2.01	2.010MHz	HC33	1.95
CY2.50	2.500MHz	HC33	4.95
CY3.27	3.2768MHz	HC33	4.95
CY3.57	3.579545MHz	HC33	4.95
CY3A	4.000MHz	HC18	4.95
CY4.91	4.916MHz	HC18	4.95
CY7A	5.000MHz	HC18	4.95
CY5.18	5.185MHz	HC18	4.95
CY6.14	6.144MHz	HC18	4.95
CY6.40	6.400MHz	HC18	4.95
CY6.55	6.5536MHz	HC18	4.95
CY12A	10.000MHz	HC18	4.95
CY14A	14.31818MHz	HC18	4.95
CY19A	18.000MHz	HC18	4.95
CY18.43	18.432MHz	HC18	4.95
CY22A	20.000MHz	HC18	4.95
CY30A	32.000MHz	HC18	4.95

TRIMMERS

10MM size trimmers -.394" Dia.

Part No.	1-9	10-24	25-49	100+
TR-11(value)	.35	.30	.25	20

Resistance values - 100, 500, 1K, 2K, 5K, 10K, 20K, 50K, 100K, 200K, 1 meg

TRIMPOTS

Single-Turn - 1/2 Watt

Square - Top Adjust - 3/8" Size

Part No.	1-9	10-24	25-49	50-99
63P(value)	.99	.89	.80	.70

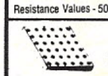
Resistance Values - 50, 100, 500, 1K, 2K, 5K, 10K, 20K, 50K, 100K, 200K, 500K, 1 meg.

15-Turn - 3/4 Watt

Rectangular Side Adjust 3/4" x 1/4" Size

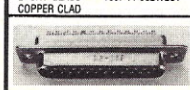
Part No.	1-9	10-24	25-49	50-99
43P(value)	1.35	1.25	1.20	1.15

Resistance Values - 50, 100, 500, 1K, 2K, 5K, 10K, 20K, 50K, 100K, 200K, 500K, 1 meg.



1/16 VECTOR BOARD

	0.1" Hole Spacing Part No.	P-Pattern L	W	Price 1-9	10 up
PHENOLIC	64P44 062XXXP	4.50	6.50	1.72	1.54
	169P44 062XXXP	4.50	17.00	3.69	3.32
EPOXY	64P44 062WE	4.50	6.50	2.07	1.86
GLASS	84P44 062WE	4.50	8.50	2.56	2.31
	169P44 062WE	4.50	17.00	5.04	4.53
	169P84 062WE	8.50	17.00	9.23	8.26
EPOXY GLASS COPPER CLAD	169P44 062WEC1	4.50	17.00	6.80	6.12



CONNECTORS

25 Pin-D Subminiature

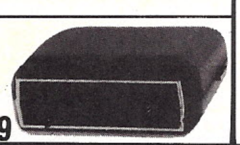
DB25P(as pictured)	PLUG	$3.25
DB25S	SOCKET	4.95
DB51226-1	Cover for DB25 P or S	1.75

MOLEX CONNECTOR PINS

M-530-1 $1.95/100 pins (minimum order)
$16.00/1000 pins

Pre-packaged in strips

INSTRUMENT/ CLOCK CASE



Injection molded unit.
Complete with red bezel.
4½" x 4" x 1-9/16".

$3.49

MICROPROCESSOR COMPONENTS

P8085	CPU	$29.95	CDP 1802	CPU	$19.95
8080A	CPU	10.95	Z80	CPU	24.95
8212	8-Bit Input/Output	4.95	2650	MPU	26.50
8214	Priority Interrupt Control	7.95	MC6800	MPU	19.95
8216	Bi-Directional Bus Driver	4.95	MC6810API	128 x 8 Static Ram	5.95
8224	Clock Generator/Driver	5.95	MC6820	Periph. Interface Adapter	7.95
8228	System Controller/Bus Driver	5.95	MC6821	Periph. Interface Adapter	11.50
8251	Prog. Comm. Interface	9.95	MC6830L8	1024 x 8 Bit ROM	14.95
8255	Prog. Periph. Interface	10.95	MC6850	Asychronous Comm. Adapter	14.95

RAM'S

1101	256 x 1	Static	$ 1.49
1103	1024 x 1	Dynamic	.99
2101	256 x 4	Static	5.95
2102	1024 x 1	Static	1.75
2107/5280	4096 x 1	Dynamic	4.95
2111	256 x 4	Static	6.95
2112	256 x 4	Static	5.95
2114	4K x 1	Static 450ns	9.95
2114L	4K x 1	Static 450ns Low Power	10.95
2114-3	1K x 4	Static 300ns	10.95
2114L-3	1K x 4	Static 300ns Low Power	11.95
7489	16 x 4	Static	1.75
8101	256 x 4	Static	5.95
8111	256 x 4	Static	6.95
8599	16 x 4	Static	3.49
21L02	1024 x 1	Static	1.95
74200	256 x 1	Static	6.95
93421	256 x 1	Static	2.95
MM5262	2K x 1	Dynamic	3/1.00
MK4027 (UPD414)	4K	DYNAMIC 16 PIN	5.95
MK4116 (UPD416)	16K	DYNAMIC 16 PIN	29.95
TMS4044-45NL	4K	STATIC	14.95

ROM'S

2513(2140)	Character Generator (upper case)	$ 9.95
2513(3021)	Character Generator (lower case)	9.95
2516	Character Generator	10.95
MM5230N	2048 Bit Read Only Memory	1.95

USER MANUALS

1802M	CDP1802 Manual	$ 7.50
Z80M	Z80 Manual	7.50
2650M	2650 Manual	5.00

PROMS

1702A	2048 x 1	Famous	$ 5.95
5203	2048 x 1	Famous	14.95
82S23	32 x 8	Open C	5.00
82S115	4096 x 1	Bipolar	19.95
82S123	32 x 8	Tristate	5.00
74S287	1024 x 1	Static	7.95
2708	8K	EPROM	10.95
2716 T.I.	16K	EPROM	29.95
2716 Intel	16K	EPROM	59.95
6301-1	1024 x 1	Tri-State Bipolar	3.49
6330-1	256 x 1	Open C Bipolar	2.95
74186	512 x 1	TTL Open Collector	9.95
74188	256 x 1	TTL Open Collector	3.95

SHIFT REGISTERS

MM5013N	1024 BitAccumulator Dynamic	2.95
MM5016H	500/512 Bit Dynamic	.89
MM5017N	Dual 500/512 Bit Dynamic	2.95
2504T	1024 Dynamic	3.95
2518	Hex 32 Bit Static	4.95
2519	Hex 40 Bit Static	4.00
2522	Dual 132 Bit Static	2.95
2524	512 Dynamic	.99
2525	1024 Dynamic	2.95
2527	Dual 256 Bit Static	2.95
2528	Dual 250 Static	4.00
2529	Dual 240 Bit Static	4.00
2532	Quad 80 Bit Static	2.95
2533	1024 Static	2.95
3341	Fifo	6.95
74LS670	4 X 4 Register	1.95

UART'S

AY-5-1013	30K BAUD	$ 5.95

SPECIAL REQUESTED ITEMS

TELEPHONE KEYBOARD CHIPS

AY-5-9100	$14.95
AY-5-9200	14.95
AY-5-9500	4.95
AY-5-2376	14.95
HD0165	7.95
74C922	9.95

ICM CHIPS

ICM7045	$24.95
ICM7205	19.95
ICM7207	7.50
ICM7208	19.95
ICM7209	6.95

NMOS READ ONLY MEMORIES

MCM6571	$13.50
MCM6574	13.50
MCM6575	13.50

TV GAME CHIP SET
AY-3-8500-1 Chip and 2.010 MH2 Crystal **$7.95**

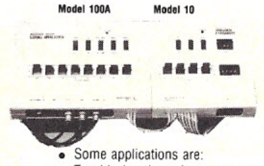
MISCELLANEOUS

11C90	$19.95	MK40240	$17.50
MC3061P	11.95	DS0026CH	3.75
MC1408L7	4.95	TIL308	10.20
MC1408L8	5.75	95H90	11.95
LD110/111	$25.00/set		
MC4016(74416)	7.50		
4N33	3.95		

PARATRONICS

Logic Analyzer Kit Model 100A

$229.00/kit



- Analyzes any type of digital system
- Checks data rates in excess of 8 million words per second
- Trouble shoot TTL, CMOS, DTL, RTL, Schottky and MOS families
- Displays 16 logic states up to 8 digits wide
- See ones and zeros displayed on your CRT, octal or hexadecimal format
- Tests circuits under actual operating conditions
- Easy to assemble — comes with step-by-step construction manual which includes 80 pages on logic analyzer operation.

(Model 100A Manual - $4.95)

- Some applications are:
  - — Troubleshooting microprocessor address, instruction, and data flow
  - — Examine contents of ROMS
  - — Tracing operation of control logic
  - — Checking counter and shift register operation
  - — Monitoring I/O sequences
  - — Verifying proper system operations during testing

PARATRONICS TRIGGER EXPANDER - Model 10

Adds 16 additional bits. Provides digital delay and qualification of input clock and 24-bit trigger word. — Connects direct to Model 100A for integrated unit).

Model 10 Kit - $229.00
Baseplate — $9.95
Model 10 Manual — $4.95

B+K PRECISION

3½-Digit Portable DMM

- Overload Protected
- .3" high LED Display
- Battery or AC operation
- Auto Zeroing
- 1mv, 1Va, 0.1 ohm resolution
- Overange reading
- 10 meg input impendence
- DC Accuracy 1% typical

Ranges: DC Voltage ±0-1000V/
AC Voltage: 0-1000V
Freq. Response: 50-400 HZ
DC/AC Current: 0-100mA
Resistance 0-10 meg ohm
Size: 6.4" x 4.4" x 2"

Model 2800
$99.95

Comes with test leads, operating manual and spare fuse.

Accessories:

AC Adapter BC-28	**$9.00**
Rechargeable Batteries BP-26	**20.00**
Carrying Case LC-28	**7.50**

CSC
CONTINENTAL SPECIALTIES CORPORATION

100 MHz 8-Digit Counter

- 20 Hz-100 MHz Range
- .6" LED Display
- Crystal-controlled timebase
- Fully Automatic
- Portable — completely self-contained
- Size — 1.75" x 7.38" x 5.63"
- Four power souces, i.e. batteries, 110 or 220V with charger 12V with auto lighter adapter and external 7.2-10V power supply

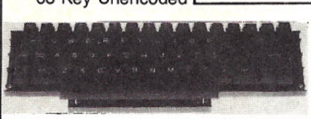
MAX-100 $134.95

ACCESSORIES FOR MAX 100:

Mobile Charger Eliminator
use power from car battery — **Model 100 — CLA $3.95**

Charger/Eliminator
use 110 V AC — **Model 100 — CAI $9.95**

63-Key Unencoded **KEYBOARDS** Hexadecimal Encoder



This is a 63-key, terminal keyboard newly manufactured by a large computer manufacturer. It is unencoded with SPST keys, unattached to any kind of PC board. A very solid molded plastic 13 x 4" base suits most application. IN STOCK **$29.95/each**

19-key pad includes 1-10 keys, ABCDEF and 2 optional keys and a shift key. **$10.95/each**

The Incredible "Pennywhistle 103"

$129.95 Kit Only

The Pennywhistle 103 is capable of recording data to and from audio tape without critical speed requirements for the recorder and it is able to communicate directly with another modem and terminal for telephone "hamming" and communications for the deaf. In addition, it is free of critical adjustments and is built with non-precision, readily available parts.

Data Transmission Method .....Frequency-Shift Keying, full-duplex (half-duplex selectable).
Maximum Data Rate...........300 Baud.
Data Format ....Asynchronous Serial (return to mark level required between each character).
Receive Channel Frequencies ...2025 Hz for space; 2225 Hz for mark.
Transmit Channel Frequencies ..Switch selectable: Low (normal) = 1070 space, 1270 mark; High = 025 space, 2225 mark.
Receive Sensitivity............−46 dbm accoustically coupled.
Transmit Level ....−15 dbm nominal. Adjustable from −6 dbm to −20 dbm.
Receive Frequency Tolerance ...Frequency reference automatically adjusts to allow for operation between 1800 Hz and 2400 Hz.
Digital Data Interface..........EIA RS-232C or 20 mA current loop (receiver is optoisolated and non-polar).
Power Requirements ....120 VAC, single phase, 10 Watts.
Physical.......................All components mount on a single 5" by 9" printed circuit board. All components included.

Requires a VOM, Audio Oscillator, Frequency Counter and/or Oscilloscope to align.

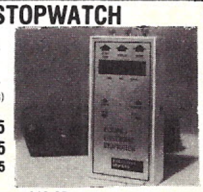
The Original the 3rd Hand

$9.95 each

- *Leaves two hands free for working
- * Clamps on edge of bench, table or work bench
- * Position board on angle or flat position for soldering or clipping
- * Sturdy, aluminum construction for hobbyist, manufacturer or school rooms

DIGITAL STOPWATCH

- Bright 6 Digit LED Display
- Times to 59 minutes 59.59 seconds
- Crystal Controlled Time Base
- Three Stopwatches in One
  Times Single Event — Split & Taylor
- Size 4.5" x 2.15" x .90" (4½ ounces)
- Uses 3 Penlite Cells.



Kit — $39.95
Assembled — $49.95
Heavy Duty Carry Case $5.95

Stop Watch Chip Only (7205) $19.95

IMC 3½ DIGIT DPM KIT

- New Bipolar Unit
- Auto Zeroing
- .5" LED
- Auto Polarity
- Low Power
- Single IC Unit

Model KB500 DPM Kit	**$49.00**
Model KB503 5V Power Kit	**$17.50**

JE700 CLOCK

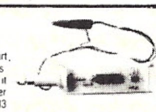
The JE700 is a low cost digital clock, but is a very high quality unit. The unit features a simulated walnut case with dimensions of 6" x 2½" x 1". It utilizes a MAN72 high brightness readout, and the MM5314 clock chip.

12 or 24 Hour

115 VAC — *KIT ONLY* **$16.95**

JE803 PROBE

The Logic Probe is a unit which is for the most part, indespensible in trouble shooting logic families TTL, DTL, RTL, CMOS. It derives the power it needs to operate directly off of the circuit under test, drawing a scant 10 mA max. It uses a MAN3 readout to indicate any of the following states by these symbols (H) - 1 (LOW) - o (PULSE) - P. The Probe can detect high frequency pulses to 45 MHz. It can't be used at MOS levels or circuit damage will result.



$9.95 Per Kit

printed circuit board

T^2L 5V 1A Supply

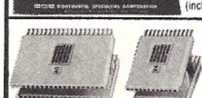
This is a standard TTL power supply using the well known LM309K regulator IC to provide a solid 1 AMP of current at 5 volts. We try to make things easy for you by providing everything you need in one package, including the hardware for only **JE225 $9.95 Per Kit**

CSC
CONTINENTAL SPECIALTIES CORPORATION

PROTO BOARDS

PROTO BOARD 6
$15.95
(6" long X 4" wide)

PB100 - 4.5" x 6"	$ 19.95
PB101 - 5.8" x 4.5"	29.95
PB102 - 7" x 4.5"	39.95
PB103 - 9" x 6"	59.95
PB104 - 9.5" x 8"	79.95
PB203 - 9.75 x 6½ x 2¾	80.00
PB203A - 9.75 x 6½ x 2¾ (includes power supply)	129.95



PROTO CLIPS

14 PIN	$4.50
16 PIN	4.75
24 PIN	8.50
40 PIN	13.75

$5.00 Minimum Order – U.S. Funds Only
California Residents – Add 6% Sales Tax

Spec Sheets – 25¢
1978A Catalog Available–Send 35¢ stamp

NEW NAME

Jameco ELECTRONICS

a division of JAMES ELECTRONICS of California

PHONE ORDERS WELCOME (415) 592-8097

MAIL ORDER ELECTRONICS – WORLDWIDE
1021 HOWARD AVENUE, SAN CARLOS, CA 94070
Advertised Prices Good Thru August

7400 TTL

SN7400N	.16				
SN7401N	.18	SN7472N	.29	SN74160N	.89
SN7402N	.18	SN7473N	.35	SN74161N	.89
SN7403N	.18	SN7474N	.35	SN74162N	1.95
SN7404N	.18	SN7475N	.49	SN74163N	.89
SN7405N	.20	SN7476N	.35	SN74164N	.89
SN7406N	.29	SN7479N	5.00	SN74165N	.89
SN7407N	.29	SN7480N	.50	SN74166N	1.25
SN7408N	.20	SN7482N	.99	SN74167N	1.95
SN7409N	.20	SN7483N	.59	SN74170N	1.59
SN7410N	.18	SN7485N	.79	SN74172N	6.00
SN7411N	.25	SN7486N	.35	SN74173N	1.25
SN7412N	.25	SN7489N	1.75	SN74174N	.89
SN7413N	.40	SN7490N	.45	SN74175N	.79
SN7414N	.70	SN7491N	.59	SN74176N	.79
SN7416N	.25	SN7492N	.43	SN74177N	.79
SN7417N	.25	SN7493N	.43	SN74179N	1.95
SN7420N	.20	SN7494N	.65	SN74180N	.79
SN7421N	.29	SN7495N	.65	SN74181N	1.95
SN7422N	.39	SN7496N	.65	SN74182N	.79
SN7423N	.25	SN7497N	3.00	SN74184N	1.95
SN7425N	.29	SN74100N	.89	SN74185N	1.95
SN7426N	.29	SN74107N	.35	SN74186N	9.95
SN7427N	.25	SN74109N	.59	SN74188N	3.95
SN7429N	.39	SN74116N	1.95	SN74190N	1.25
SN7430N	.20	SN74121N	.35	SN74191N	1.25
SN7432N	.25	SN74122N	.39	SN74192N	.79
SN7437N	.25	SN74123N	.49	SN74193N	.79
SN7438N	.25	SN74125N	.49	SN74194N	.89
SN7439N	.25	SN74126N	.49	SN74195N	.69
SN7440N	.20	SN74132N	.75	SN74196N	.89
SN7441N	.89	SN74136N	.75	SN74197N	.89
SN7442N	.49	SN74141N	.79	SN74198N	1.49
SN7443N	.75	SN74142N	2.95	SN74199N	1.49
SN7444N	.75	SN74143N	2.95	SN74200N	5.59
SN7445N	.75	SN74144N	2.95	SN74251N	1.79
SN7446N	.69	SN74145N	.79	SN74279N	.79
SN7447N	.59	SN74147N	1.95	SN74283N	2.25
SN7448N	.79	SN74148N	1.29	SN74284N	3.95
SN7450N	.20	SN74150N	.89	SN74285N	3.95
SN7451N	.20	SN74151N	.59	SN74365N	.69
SN7453N	.20	SN74153N	.59	SN74366N	.69
SN7454N	.20	SN74154N	.99	SN74367N	.69
SN7459A	.25	SN74155N	.79	SN74368N	.69
SN7460N	.20	SN74156N	.79	SN74390N	1.95
SN7470N	.29	SN74157N	.65	SN74393N	1.95

C/MOS

CD4000	.23			CD4070	.55
CD4001	.23			CD4071	.23
CD4002	.23	CD4028	.89	CD4072	.49
CD4006	1.19	CD4029	1.19	CD4076	1.39
CD4007	.25	CD4030	.49	CD4081	.23
CD4009	.49	CD4035	.99	CD4082	.23
CD4010	.49	CD4040	1.19	CD4093	.99
CD4011	.23	CD4041	1.25	CD4098	2.49
CD4012	.25	CD4042	.99	MC14409	14.95
CD4013	.39	CD4043	.89	MC14410	14.95
CD4014	1.39	CD4044	.89	MC14411	14.95
CD4015	1.19	CD4046	1.79	MC14419	4.95
CD4016	.49	CD4047	2.50	MC14433	19.95
CD4017	1.19	CD4048	1.35	MC14506	.75
CD4018	.99	CD4049	.49	MC14507	.99
CD4019	.49	CD4050	.49	MC14562	14.50
CD4020	1.19	CD4051	1.19	MC14583	3.50
CD4021	1.39	CD4053	1.19	CD4508	3.95
CD4022	1.19	CD4056	2.95	CD4510	1.39
CD4023	.23	CD4059	9.95	CD4511	1.29
CD4024	.79	CD4060	1.49	CD4515	2.95
CD4025	.23	CD4066	.79	CD4518	1.29
CD4026	2.25	CD4068	.39	CD4520	1.29
CD4027	.69	CD4069	.45	CD4566	2.25

74C00

74C00	.39			74C163	3.00
74C02	.55			74C164	3.25
74C04	.75	74C89	6.49	74C173	2.60
74C08	.75	74C90	3.00	74C192	3.49
74C10	.65	74C93	2.00	74C193	2.75
74C14	3.00	74C95	2.00	74C195	2.75
74C20	.65	74C107	1.25	74C922	9.95
74C30	.65	74C151	2.90	74C923	8.95
74C42	2.15	74C154	3.00	74C925	14.95
74C48	4.75	74C157	2.15	74C926	11.95
74C73	1.50	74C160	3.25	80C95	1.50
74C74	1.15	74C161	3.25	80C97	1.50

LINEAR

78MG	1.75			LM733N	1.00
LM300H	.80			LM739N	1.19
LM301H	.35			LM741CH	.35
LM301CN	.35			LM741CN	.35
LM302H	.75	LM340T-8	1.25	LM741-14N	.39
LM304H	1.00	LM340T-12	1.25	LM747H	.79
LM305H	.60	LM340T-15	1.25	LM747N	.79
LM307CN/H	.35	LM340T-18	1.25	LM748H	.39
LM308H	1.00	LM340T-24	1.25	LM748N	.39
LM308CN	1.00	LM350N	1.00	LM1303N	.89
LM309H	1.10	LM351CN	.65	LM1304N	1.19
LM309K	1.25	LM370N	1.15	LM1305N	1.40
LM310CN	1.15	LM373N	3.25	LM1307N	.85
LM311H	.90	LM377N	4.00	LM1310N	2.95
LM311N	.90	LM380N	1.25	LM1351N	1.65
LM317K	6.50	LM380CN	.99	LM1414N	1.75
LM318CN	1.50	LM381N	1.79	LM1458CN/H	.59
LM319N	1.30	LM382N	1.79	MC1488	1.95
LM320K-5	1.35	NE501N	8.00	MC1489	1.95
LM320K-5.2	1.35	NE510A	6.00	LM1496N	.95
LM320K-12	1.35	NE529A	4.95	LM1556V	1.75
LM320K-15	1.35	NE531H	3.00	MC1741SCP	3.00
LM320K-18	1.35	NE536T	6.00	LM2901N	2.95
LM320K-24	1.35	NE540L	6.00	LM3053	1.50
LM320T-5	1.25	NE550N	1.30	LM3065N	.69
LM320T-5.2	1.25	NE555V	.39	LM3900N(3401)	.49
LM320T-8	1.25	NE556	.99	LM3905N	.89
LM320T-12	1.25	NE5608	5.00	LM3909	1.25
LM320T-15	1.25	NE561B	5.00	MC5558V	1.00
LM320T-18	1.25	NE562B	5.00	LM7525N	.90
LM320T-24	1.25	NE565H	1.75	LM7534N	.75
LM323K-5	5.95	NE565N	1.25	80388	4.95
LM324N	1.80	NE566CN	1.75	LM75450	.49
LM339N	.99	NE567H	1.25	75451CN	.39
LM340K-5	1.35	NE567V	.99	75452CN	.39
LM340K-6	1.35	NE570	10.50	75453CN	.39
LM340K-8	1.35	LM703CN/H	.45	75454CN	.39
LM340K-12	1.35	LM709H	.29	75491CN	.79
LM340K-15	1.35	LM709N	.29	75492CN	.89
LM340K-18	1.35	LM710N	.79	75494CN	.89
LM340K-24	1.35	LM711N	.39	RC4151	5.95
LM340T-5	1.25	LM723H	.55	RC4194	5.95
LM340T-6	1.25	LM723N	.55	RC4195	4.49

74LS00 TTL

74LS00	.23			74LS139	.69
74LS01	.23			74LS151	.69
74LS02	.23	74LS51	.23	74LS155	.69
74LS03	.23	74LS54	.23	74LS157	.69
74LS04	.29	74LS55	.23	74LS160	.89
74LS05	.29	74LS73	.35	74LS161	.89
74LS08	.23	74LS74	.35	74LS162	.89
74LS09	.29	74LS75	.49	74LS163	.89
74LS10	.23	74LS76	.35	74LS164	.99
74LS13	.49	74LS83	.75	74LS175	.79
74LS14	.99	74LS85	.99	74LS181	2.49
74LS15	.29	74LS86	.35	74LS190	.89
74LS20	.23	74LS90	.49	74LS191	.89
74LS21	.29	74LS92	.59	74LS192	.89
74LS22	.29	74LS93	.59	74LS193	.89
74LS26	.29	74LS95	.79	74LS194	.89
74LS27	.29	74LS96	.89	74LS195	.89
74LS28	.29	74LS107	.35	74LS253	.79
74LS30	.23	74LS109	.35	74LS257	.69
74LS32	.29	74LS112	.35	74LS260	.55
74LS37	.35	74LS123	.99	74LS279	.59
74LS40	.29	74LS132	.79	74LS367	.59
74LS42	.69	74LS136	.39	74LS368	.59
74LS47	.69	74LS138	.69	74LS670	1.95

BUGBOOK®
Continuing Education Series

BUGBOOKS I and II — **$17.00 per set**
by Peter R. Rony, David G. Larsen, WB4HYJ
Sold as a set these two books outline over 90 experiments designed to teach the reader all he will need to know about TTL logic chips to use them in conjunction with microprocessor systems. You'll learn about the basic concepts of digital electronics including gates, flip-flops, latches, buses, decoders, multiplexers, demultiplexers, LED displays, RAM's, ROM's, and much, much more.

BUGBOOK IIa — **$5.00**
by Peter R. Rony, David G. Larsen, WB4HYJ
This volume will introduce you to the fabulous UART chip — that all important interface between data terminals, etc., and your microcomputer. It also covers current loops, and the RS 232C interface standard. Particularly recommended for any RTTY enthusiast.

BUGBOOK III — **$15.00**
by Peter R. Rony, David G. Larsen, WB4HYJ, Jonathan A. Titus
Here is the book that puts it all together. Besides having much valuable text there are a series of experiments in which the reader completely explores the 8080 chip pin by pin and introduces you to the Mark 80 microcomputer, a unique easily interfaced system. It is recommended that you have the background on the BUGBOOKS I & II before proceeding with BUGBOOK III.

BUGBOOK V and VI — **$19.00 per set**
by David G. Larsen, Peter R. Rony, Jonathan A. Titus
Experiments in digital electronics, 8080A microcomputer programming and 8080A microcomputer interfacing. An integrated approach to self-instructed basic digital electronics, breadboarding and 8080A interfacing/programming. Bugbook VI integrates the digital concepts of Bugbook V into a treatment of 8080A microcomputer programming and interfacing. Detail & laboratory experiments included with each book.

CMOS-M — DESIGNERS PRIMER AND HANDBOOK New expanded version — **$8.50**
Starts at basic structure of CMOS devices through integration into MSI

THE 555 TIMER APPLICATIONS SOURCEBOOK WITH EXPERIMENTS — **$6.95**
by Howard M. Berlin W3HB
This book shows you what the 555 timer is and how to use it. Included are over 100 various design techniques, equations and graphs to create "ready-to-go" timers, generators, power supplies, measurement and control circuits, party games, circuits for the home and automobile, photography, music and Amateur Radio.

INSTRUCTOR'S MANUAL — **$3.00**
Necessary for instruction of Bugbook I and II. Answers questions regarding experiments, suggestions for further reading, philosophy of authors approach to digital electronics. A must for self-teaching individuals.

OP AMP MANUAL by Howard M. Berlin W3HB — **$9.00**
An experiment guide to application of operational amplifiers. Over 25 experiments on all phases of Op Amps.

DBUG — **$5.00**
8080 interpretive debugger. A program for entering, debugging and storing assembly language programs.

COMPLETE MANUAL FOR DIGITAL CLOCKS by John Weiss and John Brooks
Familiarizes technician or hobbyist with basic theories behind digital clocks. Includes trouble shooting guides, basic characteristics of clocks, soldering techniques, clock component data sheets and construction tips. **$3.95**

DISCRETE LEDS

.125" dia.

XC209	Red	5/$1
XC209	Green	4/$1
XC209	Orange	4/$1
XC209	Yellow	4/$1

.200" dia

XC22	Red	5/$1
XC22	Green	4/$1
XC22	Yellow	4/$1
XC22	Orange	4/$1
SSL-22	RT	4/$1

.185" dia.

XC526	Red	5/$1
XC526	Red	100/$8
XC526	Green	4/$1
XC526	Yellow	4/$1
XC526	Clear	4/$1

.200" dia.

XC556	Red	5/$1
XC556	Red	100/$8
XC556	Green	4/$1
XC556	Yellow	4/$1
XC556	Orange	4/$1
XC556	Clear	7/$1

.190" dia.

XC111	Red	5/$1
XC111	Green	4/$1
XC111	Yellow	4/$1
XC111	Orange	4/$1

.085" dia.
MV50 - Red - 6/$1
.170" dia.
MV10 Red - 4/$1

INFRA-RED LED ¼" x ¼" x 1/16" Flat 5/$1.00

DISPLAY LEDS

TYPE	POLARITY	HT	PRICE	TYPE	POLARITY	HT	PRICE
MAN 1	Common Anode-red	.270	2.95	MAN 6680	Common Cathode-orange	.560	.99
MAN 2	5 x 7 Dot Matrix-red	.300	4.95	MAN 6710	Common Anode-red-D.D.	.560	.99
MAN 3	Common Cathode-red	.125	.25	MAN 6730	Common Anode-red ±1	.560	.99
MAN 4	Common Cathode-red	.187	1.95	MAN 6740	Common Cathode-red-D.D.	.560	.99
MAN 52	Common Anode-green	.300	1.25	MAN 6750	Common Cathode-red ±1	.560	.99
MAN 71	Common Anode-red	.300	1.25	MAN 6760	Common Anode-red	.560	.99
MAN 72	Common Anode-red	.300	.99	MAN 6780	Common Cathode-red	.560	.99
MAN 74	Common Cathode-red	.300	1.25	DL701	Common Anode-red ±1	.300	.99
MAN 81	Common Anode-yellow	.300	.99	DL702	Common Cathode-red	.300	1.25
MAN 82	Common Anode-yellow	.300	.99	DL704	Common Cathode-red	.300	.99
MAN 84	Common Cathode-yellow	.300	.99	DL707	Common Anode-red	.300	.99
MAN 3620	Common Anode-orange	.300	.99	DL741	Common Anode-red	.600	1.25
MAN 3630	Common Anode-orange ±1	.300	.99	DL746	Common Anode-red ±1	.630	1.49
MAN 3640	Common Cathode-orange	.300	.99	DL747	Common Anode-red	.600	1.49
MAN 4610	Common Anode-orange	.300	.99	DL749	Common Cathode-red ±1	.630	1.49
MAN 4640	Common Cathode-orange	.400	.99	DL750	Common Cathode-red	.600	1.49
MAN 4710	Common Anode-red ±1	.400	.99	DL338	Common Cathode-red	.110	.35
MAN 4730	Common Anode-red	.400	.99	FND70	Common Cathode	.250	.69
MAN 4740	Common Cathode-red	.400	.99	FND359	Common Anode	.350	.75
MAN 4810	Common Anode-yellow	.400	.99	FND503	Common Cathode (FND500)	.500	.99
MAN 6610	Common Anode-orange-D.D.	.560	.99	FND507	Common Anode (FND510)	.500	.99
MAN 6630	Common Anode-orange	.560	.99	5082-7300	4 x 7 Sgl. Digit-RHDP	.600	19.95
MAN 6640	Common Cathode-orange-D.D.	.560	.99	5082-7302	4 x 7 Sgl. Digit-LHDP	.600	19.95
MAN 6650	Common Cathode-orange ±1	.560	.99	5082-7304	Overrange character (±1)	.600	15.00
MAN 6660	Common Anode-orange	.560	.99	5082-7340	4 x 7 Sgl. Digit-Hexadecimal	.600	22.50

RCA LINEAR

CA3013	2.15	CA3082	2.00
CA2023	2.56	CA3083	1.60
CA3035	2.48	CA3086	.85
CA3039	1.35	CA3089	3.75
CA3046	1.30	CA3130	1.39
CA3059	3.25	CA3140	1.25
CA3060	3.25	CA3160	1.25
CA3080	.85	CA3401	.49
CA3081	2.00	CA3600	3.50

CALCULATOR CHIPS AND DRIVER

FCM3817	$ 5.00
MM5725	2.95
MM5736	1.95
MM5738	2.95
DM8864	2.00
DM8865	1.00
DM8887	.75
DM8889	.75
5030	7.95

CLOCK CHIPS

MM5309	$ 9.95
MM5311	4.95
MM5312	4.95
MM5314	4.95
MM5316	6.95
MM5318	9.95
MM5369	2.95
MM5841	9.95
7001	5.95

9374 - 7-segment LED driver common anode LEDs **$ .99**

IC SOLDERTAIL — LOW PROFILE (TIN) SOCKETS

	1-24	25-49	50-100		1-24	25-49	50-100
8 pin LP	.$17	.16	.15	22 pin LP	$ .37	.36	.35
14 pin LP	.20	.19	.18	24 pin LP	.38	.37	.36
16 pin LP	.22	.21	.20	28 pin LP	.45	.44	.43
18 pin LP	.29	.28	.27	36 pin LP	.60	.59	.58
20 pin LP	.34	.32	.30	40 pin LP	.63	.62	.61

SOLDERTAIL STANDARD (TIN)

14 pin ST	$.27	.25	.24	28 pin ST	$ .99	.90	.81
16 pin ST	.30	.27	.25	36 pin ST	1.39	1.26	1.15
18 pin ST	.35	.32	.30	40 pin ST	1.59	1.45	1.30
24 pin ST	.49	.45	.42				

SOLDERTAIL STANDARD (GOLD)

8 pin SG	$.30	.27	.24	24 pin SG	$ .70	.63	.57
14 pin SG	.35	.32	.29	28 pin SG	1.10	1.00	.90
16 pin SG	.38	.35	.32	36 pin SG	1.75	1.40	1.26
18 pin SG	.52	.47	.43	40 pin SG	1.75	1.59	1.45

WIRE WRAP SOCKETS (GOLD) LEVEL #3

8 pin WW	$.40	.38	.35	22 pin WW	$ .95	.85	.75
10 pin WW	.45	.41	.37	24 pin WW	1.05	.95	.85
14 pin WW	.39	.38	.37	28 pin WW	1.40	1.25	1.10
16 pin WW	.43	.42	.41	36 pin WW	1.59	1.45	1.30
18 pin WW	.75	.68	.62	40 pin WW	1.75	1.55	1.40

50 PCS. RESISTOR ASSORTMENTS $1.75 PER ASST.

		10 OHM	12 OHM	15 OHM	18 OHM	22 OHM		
ASST. 1	5 ea.	27 OHM	33 OHM	39 OHM	47 OHM	56 OHM	1/4 WATT 5%	50 PCS.
		68 OHM	82 OHM	100 OHM	120 OHM	150 OHM		
ASST. 2	5 ea.	180 OHM	220 OHM	270 OHM	330 OHM	390 OHM	1/4 WATT 5%	50 PCS.
		470 OHM	560 OHM	680 OHM	820 OHM	1K		
ASST. 3	5 ea.	1.2K	1.5K	1.8K	2.2K	2.7K	1/4 WATT 5%	50 PCS.
		3.3K	3.9K	4.7K	5.6K	6.8K		
ASST. 4	5 ea.	8.2K	10K	12K	15K	18K	1/4 WATT 5%	50 PCS.
		22K	27K	33K	39K	47K		
ASST. 5	5 ea.	56K	68K	82K	100K	120K	1/4 WATT 5%	50 PCS.
		150K	180K	220K	270K	330K		
ASST. 6	5 ea.	390K	470K	560K	680K	820K	1/4 WATT 5%	50 PCS.
		1M	1.2M	1.5M	1.8M	2.2M		
ASST. 7	5 ea.	2.7M	3.3M	3.9M	4.7M	5.6M	1/4 WATT 5%	50 PCS.

ASST. 8R Includes Resistor Assortments 1-7 (350 PCS.) **$9.95 ea.**

$5.00 Minimum Order – U.S. Funds Only
California Residents – Add 6% Sales Tax

Spec Sheets – 25¢
1978A Catalog Available—Send 35¢ stamp

NEW NAME

Jameco ELECTRONICS
a division of JAMES ELECTRONICS of California

PHONE ORDERS WELCOME (415) 592-8097

MAIL ORDER ELECTRONICS – WORLDWIDE
1021 HOWARD AVENUE, SAN CARLOS, CA 94070
Advertised Prices Good Thru August

J1

WIRE-WRAP KIT — WK-2-W
WRAP • STRIP • UNWRAP

- Tool for 30 AWG Wire
- Roll of 50 Ft. White or Blue 30 AWG Wire
- 50 pcs. each 1", 2", 3" & 4" lengths — pre-stripped wire.

$12.95

WIRE WRAP TOOL WSU-30
WRAP • STRIP • UNWRAP - $6.95

WIRE WRAP WIRE — 30 AWG
25 ft. min. $1.25 50 ft. $1.95 100 ft. $2.95 1000 ft. $15.00
SPECIFY COLOR — White - Yellow - Red - Green - Blue - Black

WIRE DISPENSER — WD-30
- 50 ft. roll 30 AWG KYNAR wire wrap wire **$3.95 ea.**
- Cuts wire to desired length
- Strips 1" of insulation **Specify** — Blue-Yellow-White-Red

REPLACEMENT DISPENSER SPOOLS FOR WD 30
Specify blue, yellow, white or red **$1.98/spool**

EXAR

XR2206KA $14.95
Function Generator Kit (includes chip, P.C. Board and instructions)

XR2206KB $19.95
Function Generator Kit (includes all components, P.C. Board and instructions)

XR-L555 $1.50
Micro-Power version of the popular 555 Timer and directly interchangeable. Dissipates 1/15th the power and operates down to 2.7 volts. Perfect for battery operation and CMOS circuits.

XR2242CP $1.50
Precision timing circuit for generating timing pulses in **minutes, hours** and days or up to 1 year by using two. Reduces cost of time delay circuits. Basic 555 Timer with built-in 8-bit Counter.

XR205	$ 8.40	XR1489	1.39	XR2556	$ 3.20
XR210	4.40	XR1800	3.20	XR2567	2.99
XR215	4.40	XR2206	4.40	XR3403	1.25
XR320	1.55	XR2207	3.85	XR4136	1.25
XR555	.39	XR2208	5.20	XR4151	2.85
XR556	.99	XR2209	1.75	XR4194	1.45
XR567CP	.99	XR2211	5.25	XR4202	3.60
XR567CT	1.25	XR2212	4.35	XR4212	2.05
XR1310P	1.30	XR2240	3.45	XR4558	.75
XR1468CN	3.85	XR2264	4.25	XR4739	1.15
XR1488	1.39			XR4741	1.47

ZENERS — DIODES — RECTIFIERS

TYPE	VOLTS	W	PRICE	TYPE	VOLTS	W	PRICE
1N746	3.3	400m	4/1.00	1N4005	600 PIV	1 AMP	10/1.00
1N751A	5.1	400m	4/1.00	1N4006	800 PIV	1 AMP	10/1.00
1N752	5.6	400m	4/1.00	1N4007	1000 PIV	1 AMP	10/1.00
1N753	6.2	400m	4/1.00	1N3600	50	200m	6/1.00
1N754	6.8	400m	4/1.00	1N4148	75	10m	15/1.00
1N959	8.2	400m	4/1.00	1N4154	35	10m	12/1.00
1N965B	15	400m	4/1.00	1N4305	75	25m	20/1.00
1N5232	5.6	500m	28	1N4734	5.6	1w	28
1N5234	6.2	500m	28	1N4735	6.2	1w	28
1N5235	6.8	500m	28	1N4736	6.8	1w	28
1N5236	7.5	500m	28	1N4738	8.2	1w	28
1N456	25	40m	6/1.00	1N4742	12	1w	28
1N458	150	7m	6/1.00	1N4744	15	1w	28
1N485A	180	10m	5/1.00	1N1183	50 PIV	35 AMP	1.60
1N4001	50 PIV	1 AMP	12/1.00	1N1184	100 PIV	35 AMP	1.70
1N4002	100 PIV	1 AMP	12/1.00	1N1185	150 PIV	35 AMP	1.70
1N4003	200 PIV	1 AMP	12/1.00	1N1186	200 PIV	35 AMP	1.80
1N4004	400 PIV	1 AMP	12/1.00	1N1188	400 PIV	35 AMP	3.00

SCR AND FW BRIDGE RECTIFIERS

C36D	15A @ 400V	SCR(2N1849)	$1.95
C38M	35A @ 600V	SCR	1.95
2N2328	1.6A @ 300V	SCR	.50
MDA 980-1	12A @ 50V	FW BRIDGE REC.	1.95
MDA 980-3	12A @ 200V	FW BRIDGE REC.	1.95

TRANSISTORS

C106B1	.50			2N3904	4/1.00
MPSA05	.30	2N3055	.89	2N3905	4/1.00
MPSA06	5/1.00	MJE3055	1.00	2N3906	4/1.00
TIS97	6/1.00	2N3392	5/1.00	2N4013	3/1.00
TIS98	6/1.00	2N3398	5/1.00	2N4123	6/1.00
TIS133	5/1.00	PN3567	3/1.00	PN4249	4/1.00
TIS135	5/1.00	PN3568	4/1.00	PN4250	4/1.00
40409	1.75	PN3569	4/1.00	2N4400	4/1.00
40410	1.75	MPS3638A	5/1.00	2N4401	4/1.00
40673	1.75	MPS3702	5/1.00	2N4402	4/1.00
2N918	4/1.00	2N3704	5/1.00	2N4403	4/1.00
2N2219A	3/1.00	MPS3704	5/1.00	2N4409	5/1.00
2N2221A	4/1.00	2N3705	5/1.00	2N5086	4/1.00
2N2222A	5/1.00	MPS3705	5/1.00	2N5087	4/1.00
2N2369	5/1.00	2N3706	5/1.00	2N5088	4/1.00
2N2369A	4/1.00	MPS3706	5/1.00	2N5089	4/1.00
MPS2369	5/1.00	2N3707	5/1.00	2N5129	5/1.00
2N2484	4/1.00	2N3711	5/1.00	PN5134	5/1.00
2N2906	4/1.00	2N3724A	.65	PN5138	5/1.00
2N2907	5/1.00	2N3725A	1.00	2N5139	5/1.00
2N2925	5/1.00	2N3772	2.25	2N5210	5/1.00
MJE2955	1.25	2N3823	1.00	2N5449	3/1.00
2N3053	2/1.00	2N3903	4/1.00	2N5951	3/1.00

CAPACITOR CORNER
50 VOLT CERAMIC DISC CAPACITORS

	1-9	10-99	100+		1-9	10-99	100+
10 pf	.05	.04	.03	.001µF	.05	.04	.035
22 pf	.05	.04	.03	.0047µF	.05	.04	.035
47 pf	.05	.04	.03	.01µF	.05	.04	.035
100 pf	.05	.04	.03	.022µF	.06	.05	.04
220 pf	.05	.04	.03	.047µF	.06	.05	.04
470 pf	.05	.04	.035	.1µF	.12	.09	.075

100 VOLT MYLAR FILM CAPACITORS

.001mf	.12	.10	.07	.022mf	.13	.11	.08
.0022	.12	.10	.07	.047mf	.21	.17	.13
.0047mf	.12	.10	.07	.1mf	.27	.23	.17
.01mf	.12	.10	.07	.22mf	.33	.27	.22

+20% DIPPED TANTALUMS (SOLID) CAPACITORS

.1/35V	.28	.23	.17	1.5/35V	.30	.26	.21
.15/35V	.28	.23	.17	2.2/25V	.31	.27	.22
.22/35V	.28	.23	.17	3.3/25V	.31	.27	.22
.33/35V	.28	.23	.17	4.7/25V	.32	.28	.23
.47/35V	.28	.23	.17	6.8/25V	.36	.31	.25
.68/35V	.28	.23	.17	10/25V	.40	.35	.29
1.0/35V	.28	.23	.17	15/25V	.63	.50	.40

MINIATURE ALUMINUM ELECTROLYTIC CAPACITORS

Axial Lead				Radial Lead			
.47/50V	.15	.13	.10	.47/25V	.15	.13	.10
1.0/50V	.16	.14	.11	.47/50V	.16	.14	.11
3.3/50V	.14	.12	.09	1.0/16V	.15	.13	.10
4.7/25V	.15	.13	.10	1.0/25V	.16	.14	.11
10/25V	.15	.13	.10	1.0/50V	.16	.14	.11
10/50V	.16	.14	.12	4.7/16V	.15	.13	.10
22/25V	.17	.15	.12	4.7/25V	.15	.13	.10
22/50V	.24	.20	.18	4.7/50V	.16	.14	.11
47/25V	.19	.17	.15	10/16V	.14	.12	.09
47/50V	.25	.21	.19	10/25V	.15	.13	.10
100/25V	.24	.20	.18	10/50V	.16	.14	.12
100/50V	.35	.30	.28	47/50V	.24	.21	.19
220/25V	.32	.28	.25	100/16V	.19	.15	.14
220/50V	.45	.41	.38	100/25V	.24	.20	.18
470/25V	.33	.29	.27	100/50V	.35	.30	.28
1000/16V	.55	.50	.45	220/16V	.23	.17	.16
2200/16V	.70	.62	.55	470/25V	.31	.28	.26

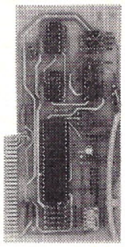
S A V E

COMPUTER INTERFACES & PERIPHERALS

S A V E

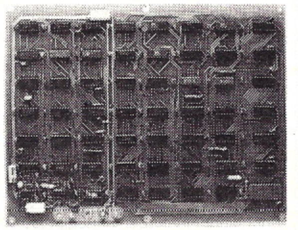
For free catalog including parts lists and schematics, send a self-addressed stamped envelope.

APPLE II SERIAL I/O INTERFACE *



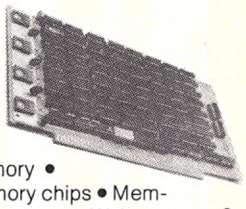
Part no. 2
• Baud rates up to 30,000 • Plugs into Apple Peripheral connector • Low-current drain • RS-232 Input and Output. SOFTWARE • Input and Output routine from monitor or BASIC to teletype or other serial printer. • Program for using an Apple II for a video or an intelligent terminal. Also can output in correspondence code to interface with some selectrics. Board only — $15.00; with parts — $42.00; assembled and tested — $62.00.

T.V. TYPEWRITER



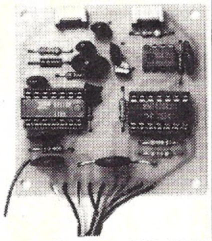
Part no. 106
• Stand alone TVT • 32 char/line, 16 lines, modifications for 64 char/line included • Parallel ASCII (TTL) input • Video output • 1K on board memory • Output for computer controlled curser • Auto scroll • Non-destructive curser • Curser inputs: up, down, left, right, home, EOL, EOS • Scroll up, down • Requires +5 volts at 1.5 amps, and -12 volts at 30 mA • All 7400, TTL chips • Char. gen. 2513 • Upper case only • Board only $39.00; with parts $145.00

8K STATIC RAM



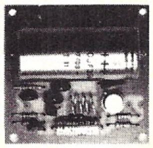
Part no. 300
• 8K Altair bus memory • Uses 2102 Static memory chips • Memory protect • Gold contacts • Wait states • On board regulator • S-100 bus compatible • Vector input option • TRI state buffered • Board only $22.50; with parts $160.00

MODEM*



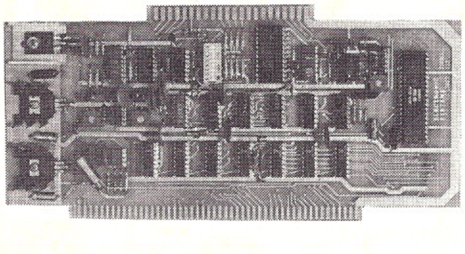
Part no. 109
• Type 103 • Full or half duplex • Works up to 300 baud • Originate or Answer • No coils, only low cost components • TTL input and output-serial • Connect 8 ohm speaker and crystal mic. directly to board • Uses XR FSK demodulator • Requires +5 volts • Board $7.60; with parts $27.50

RF MODULATOR*



Part no. 107
• Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple. • Power required is 12 volts AC C.T., or +5 volts DC • Board $7.60; with parts $13.50

TIDMA *

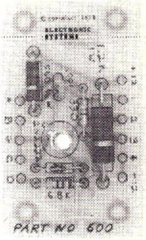


Part no. 112
• Tape Interface Direct Memory Access • Record and play programs without bootstrap loader (no prom) has FSK encoder/decoder for direct connections to low cost recorder at 1200 baud rate, and direct connections for inputs and outputs to a digital recorder at any baud rate. • S-100 bus compatible • Board only $35.00; with parts $110.00

DC POWER SUPPLY*

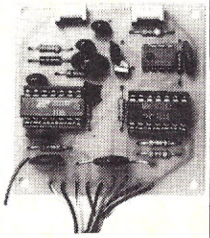
Part no. 6085
• Board supplies a regulated +5 volts at 3 amps., +12, -12, and -5 volts at 1 amp. • Power required is 8 volts AC at 3 amps., and 24 volts AC C.T. at 1.5 amps. • Board only $12.50; with parts excluding transformers $42.50

RS 232/TTY INTERFACE*



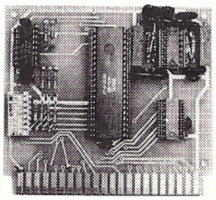
Part no. 600
• Converts RS-232 to 20mA current loop, and 20mA current loop to RS-232 • Two separate circuits • Requires +12 and -12 volts • Board only $4.50, with parts $7.00

TAPE INTERFACE *



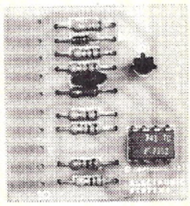
Part no. 111
• Play and record Kansas City Standard tapes • Converts a low cost tape recorder to a digital recorder • Works up to 1200 baud • Digital in and out are TTL-serial • Output of board connects to mic. in of recorder • Earphone of recorder connects to input on board • No coils • Requires +5 volts, low power drain • Board $7.60; with parts $27.50

UART & BAUD RATE GENERATOR*



Part no. 101
• Converts serial to parallel and parallel to serial • Low cost on board baud rate generator • Baud rates: 110, 150, 300, 600, 1200, and 2400 • Low power drain +5 volts and -12 volts required • TTL compatible • All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity. • All connections go to a 44 pin gold plated edge connector • Board only $12.00; with parts $35.00 with connector add $4.00

RS 232/TTL INTERFACE *



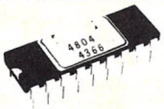
Part no. 232
• Converts TTL to RS-232, and converts RS-232 to TTL • Two separate circuits • Requires -12 and +12 volts • All connections go to a 10 pin gold plated edge connector • Board only $4.50; with parts $7.00 with connector add $3.00

ELECTRONIC SYSTEMS Dept. KB, P.O. Box 21638, San Jose, Calif. USA 95151

To Order: Mention part number and description. For parts kits add "A" to part number. In USA, shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Outside USA add 10% for air mail postage, no C.O.D.'s. Checks and money orders must be payable in US dollars. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. All items are in stock, and will be shipped the day order is received via first class mail. Prices are in US dollars. No open accounts. To eliminate tariff in Canada boxes are marked "Computer Parts." Dealer inquiries invited.
24 Hour Order Line: (408) 226-4064

*Designed by John Bell

4804 STATIC, TTL IN/OUT 1024x4 N-MOS RAM



GENERAL DESCRIPTION

Part Number 4804 is a 4K semiconductor random access memory organized as 1024 4-bit words. It is fully static and needs no clock or refresh pulses. It requires a single +5 volt power supply and is fully TTL compatible on input and output lines. The 4804 is packaged in a convenient 18 pin dual-in-line package.

FEATURES

- Single +5V Power Supply
- 1Kx4 Organization
- Replaces 4 1024x1 Static RAMs
- Completely Static—No Clocks or Refresh
- 18 Pin Package
- Access/Cycle 600nsec max
- 250 mw Typical Operating Power
- Common I/O Bus
- TTL Compatible I/O
- Three State Outputs

TRUTH TABLE

$\overline{CE}$	R/$\overline{W}$	DI/DO	STATUS	MODE
H	Don't Care	High Z	Deselect	Standby
L	H	Data	Selected	READ
L	L	L	Selected	Write 0
L	L	H	Selected	Write 1

WRITE CYCLE—AC CHARACTERISTICS

PARAMETER	SYMBOL	4804 MIN	MAX
Write Cycle Time	T_{WC}	600	
Address To Write Time	T_{AW}	100	
Write Pulse Width	T_{WP}	500	
Write Recovery Time	T_{WR}	0	
Data Set Up Time	T_{DW}	350	
Data Hold Time	T_{DH}	0	
Output Disable From Write or Chip Enable	T_{OTW}		150

READ CYCLE—AC CHARACTERISTICS

PARAMETER	SYMBOL	4804 MIN	MAX
Read Cycle Time	T_{RC}	600	
Access Time	T_A		600
Chip Enable to Output Enable	T_{CO}		200
Data Valid After Address	T_{OH1}	150	
Previous Data Valid After Chip De-Select	T_{OH2}	25	

$8.95   8/$60.00   16/$100.00

INTEGRATED TONE RECEIVER
MK5102(N)-5

FEATURES

- ☐ Detects all 16 standard DTMF digits
- ☐ Requires minimum external parts count for minimum system cost
- ☐ Uses inexpensive 3.579545 MHz crystal for reference
- ☐ Digital counter detection with period averaging insures minimum false response
- ☐ 16-pin package for high system density
- ☐ Single supply 5 Volts ± 10%
- ☐ Output in either 4-bit binary code or dual 2-bit row/column code
- ☐ Latched outputs

DESCRIPTION

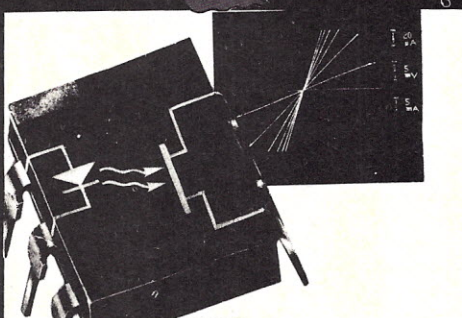
The MK5102 is a monolithic integrated circuit fabricated using the complementary-symmetry MOS (CMOS) process. Using an inexpensive 3.579545 MHz television colorburst crystal for reference, the MK5102 detects and decodes the 8 standard DTMF frequencies used in telephone dialing. The requirement of only a single supply and its construction in a 16-pin package make the MK5102 ideal for applications requiring minimum size and external parts count.

DETECTION FREQUENCY

Low Group f_o	High Group f_o
Row 1 = 697 Hz	Column 1 = 1209 Hz
Row 2 = 770 Hz	Column 2 = 1336 Hz
Row 3 = 852 Hz	Column 3 = 1477 Hz
Row 4 = 941 Hz	Column 4 = 1633 Hz

MK5102N-5............................$34.95
Specs.......................................50
600 Ohm to 600 Ohm C.T. transformer.........$1.95

Annie Sez:
If that new project has you flippin' around looking for the latest parts, come on out to sun country-try Tri-Tek.

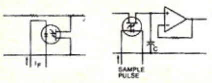


H11F3

It's GE's latest semiconductor breakthrough. The new H11F combines a dependable, efficient GaAs IRED with a Silicon Bilateral Analog FET to produce an opto-isolator with linear low level output.

A typical application of the H11F as a variable resistor yields distortion free attenuation of low level 10KHz signals over a 70 db range.

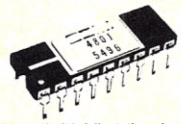
Another unique application of the H11F is in an isolated sample and hold circuit where signal polarity is undefined and cost is critical.



new from G.E.

H11F3 has 470 Ohms maximum resistance at 16 mA forward on the LED. Greater than 300 MOhm with 0 mA.
H11F3 ....6 pin DIP...$1.95   Specs...................50

4801 STATIC, TTL IN/OUT 4096x1 N-MOS RAM



GENERAL DESCRIPTION

Part Number 4801 is a 4K semiconductor random access memory organized as 4096 1-bit words. It is fully static and needs no clock or refresh pulses. It requires a single +5 volt power supply and is fully TTL compatible on input and output lines. The 4801 is packaged in a convenient 18 pin dual-in-line package.

FEATURES

- Single +5V Power Supply
- 4Kx1 Organization
- Replaces 4 1024x1 Static RAMs
- Completely Static—No Clocks or Refresh
- 18 Pin Package
- Access/Cycle Times 600 nsec max
- 250 mw Typical Operating Power
- Separate Data In and Data Out
- TTL Compatible I/O
- Three State Outputs
- Data Bus Compatible I/O Function

TRUTH TABLE

$\overline{CE}$	R/$\overline{W}$	DI	DO	STATUS	MODE
H	Don't Care	Don't Care	High Z	Deselect	Standby
L	H	Don't Care	Data	Selected	READ
L	L	L	High Z	Selected	Write 0
L	L	H	High Z	Selected	Write 1

WRITE CYCLE—AC CHARACTERISTICS

PARAMETER	SYMBOL	4801 MIN	MAX
Write Cycle Time	T_{WC}	600	
Address To Write Time	T_{AW}	100	
Write Pulse Switch	T_{WP}	500	
Write Recovery Time	T_{WR}	0	
Data Set Up Time	T_{DW}	350	
Data Hold Time	T_{DH}	0	
Output Disable From Write or Chip Enable	T_{OTW}		150

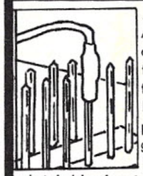
READ CYCLE—AC CHARACTERISTICS

PARAMETER	SYMBOL	4801 MIN	MAX
Read Cycle Time	T_{RC}	600	
Access Time	T_A		600
Chip Enable to Output Enable	T_{CO}		200
Data Valid After Address	T_{OH1}	150	
Previous Data Valid After Chip De-Select	T_{OH2}	25	

$8.95   8/$60.00   16/$100.00

VOLTAGE REGULATORS

7805-06-08-12-15-24 TO220	95¢ 5/$4.50
78L05A-12-15 4% 100 mA TO-92 Plastic	50¢
78H05KC 5V 5A TO-3	8.45
78H12KC 12V 5A TO-3	9.15
78H15KC 15V 5A TO-3	9.15
Lm317K 1.5A *Adjustable* TO-3	4.99
Lm317T 1.5A *Adjustable* TO-220	3.99
Lm317MP .5A *Adjustable* TO-202	13.95
TL430C Adjustable Zener-*Think About It*	1.50
TL497C Switching Reg. & Inductor	9.50
RCA CA 3085 100 mA *Adjustable*	.60
Signetics 2504TA 1024 bit S.R. memory (1404A)...	.50
MCM 6571P Character Generator	9.95
MCM6571AP Character Generator	9.95
MC14409P Telephone Rotary Pulser	10.98
MC14419P Touch Pad Converter for 14409	4.25
MC14411P Baud Rate Generator	11.98
MC14412VP CMOS Modem Chip	16.95
MM57109N Number Cruncher Micro	18.95
74C915 7 Segment to BCD Converter	2.99
74C922 16 Key Keyboard Encoder	6.35
74C923 20 key Keyboard Encoder	6.45
74C925 4 Decade Counter w/latches	12.00
74C926 4 Decade Counter w/carry	12.00
74C935-1 3½ Digit DVM CMOS Chip	16.98



Jumper Kits for .025 Square Posts.. All material for making jumpers for quick circuit changes and prototyping. Use for breadboarding, trouble shooting, field modifications. Fits standard IC socket wire/wrap posts. Excellent wiping action on gold plated box contacts.
Kit contain 10 box contacts, heat shrinkable sleeving, and 5 feet of wire plus instruction sheet.
JCK-5101....(5 double jumpers)$2.75, 4 kits/$10.00

DEALER PRICING AVAILABLE UPON REQUEST

TRI-TEK

7808 North 27th Avenue
Phoenix, Arizona 85021
(602) 995-9352

- Please give street address for UPS shipping when possible.
- C.O.D. NO parcel post C.O.D. UPS C.O.D. Add 85¢ to order.
- Any correspondence not connected with your order, please use separate sheet and include SASE for reply.
- Orders less than $10 ($15 foreign) please add $1 handling.
- Prices are subject to change without notice.
- Any refunds will be by check, not credit vouchers.
- Terms. Check, money order, credit card. Net 30 days to rated firms, schools and government agencies.
- If we should be temporarily out of stock on an item, it will be placed on back order. If we cannot ship in 30 days, you will be notified of the expected shipping date and furnished with a postage paid card with which to cancel your order if desired.
- We pay surface shipping only in USA, Canada and Mexico.
- For premium shipping (first class, special handling, etc.) add extra. Excell will be refunded.
- Foreign orders (except Canada and Mexico) estimate and add shipping. Excess will be refunded.

T1

Charge card telephone orders ($20 min.) will be accepted 9-5:30 P.M. except weekends. Telephone 995-9352. No collect calls please.

DIODES/ZENERS

1N914	100v	10mA	.05
1N4005	600v	1A	.08
1N4007	1000v	1A	.15
1N4148	75v	10mA	.05
1N4733	5.1v	1 W Zener	.25
1N753A	6.2v	500 mW Zener	.25
1N758A	10v	"	.25
1N759A	12v	"	.25
1N5243	13v	"	.25
1N5244B	14v	"	.25
1N5245B	15v	"	.25

SOCKETS/BRIDGES

8-pin	pcb	.20	ww	.35
14-pin	pcb	.20	ww	.40
16-pin	pcb	.20	ww	.40
18-pin	pcb	.25	ww	.75
22-pin	pcb	.35	ww	.95
24-pin	pcb	.35	ww	.95
28-pin	pcb	.45	ww	1.25
40-pin	pcb	.50	ww	1.25

Molex pins	.01	To-3 Sockets	.25
2 Amp Bridge		100-prv	.95
25 Amp Bridge		200-prv	1.95

TRANSISTORS, LEDS, etc.

2N2222	NPN (2N2222 Plastic .10)	.15
2N2907	PNP	.15
2N3906	PNP (Plastic - Unmarked)	.10
2N3904	NPN (Plastic - Unmarked)	.10
2N3054	NPN	.35
2N3055	NPN 15A 60v	.50
T1P125	PNP Darlington	.35
LED Green, Red, Clear, Yellow		.15
D.L.747	7 seg 5/8" High com-anode	1.95
MAN72	7 seg com-anode (Red)	1.25
MAN3610	7 seg com-anode (Orange)	1.25
MAN82A	7 seg com-anode (Yellow)	1.25
MAN74A	7 seg com-cathode (Red)	1.50
FND359	7 seg com-cathode (Red)	1.25

C MOS

4000	.15
4001	.15
4002	.20
4004	3.95
4006	.95
4007	.20
4008	.75
4009	.35
4010	.35
4011	.20
4012	.20
4013	.40
4014	.75
4015	.75
4016	.35
4017	.75
4018	.75
4019	.35
4020	.85
4021	.75
4022	.75
4023	.20
4024	.75
4025	.20
4026	1.95
4027	.35
4028	.75
4030	.35
4033	1.50
4034	2.45
4035	.75
4040	.75
4041	.69
4042	.65
4043	.50
4044	.65
4046	1.25
4049	.45
4050	.45
4066	.55
4069/74C04	.25
4071	.25
4081	.30
4082	.30
MC 14409	14.50
MC 14419	4.85
4511	.95
74C151	1.90

— T T L —

7400	.10	7473	.25	74176	.85	74H72	.35	74S133	.40		
7401	.15	7474	.30	74180	.55	74H101	.75	74S140	.55		
7402	.15	7475	.35	74181	2.25	74H103	.55	74S151	.30		
7403	.15	7476	.40	74182	.75	74H106	.95	74S153	.35		
7404	.10	7480	.55	74190	1.25			74S157	.75		
7405	.25	7481	.75	74191	.95	74L00	.25	74S158	.30		
7406	.25	7483	.75	74192	.75	74L02	.20	74S194	1.05		
7407	.55	7485	.55	74193	.85	74L03	.25	74S257 (8123)	1.05		
7408	.15	7486	.25	74194	.95	74L04	.30				
7409	.15	7489	1.05	74195	.95	74L10	.20	74LS00	.20		
7410	.15	7490	.45	74196	.95	74L20	.35	74LS01	.20		
7411	.25	7491	.70	74197	.95	74L30	.45	74LS02	.20		
7412	.25	7492	.45	74198	1.45	74L47	1.95	74LS04	.20		
7413	.25	7493	.35	74221	1.00	74L51	.45	74LS05	.25		
7414	.75	7494	.75	74367	.75	74L55	.65	74LS08	.25		
7416	.25	7495	.60			74L72	.45	74LS09	.25		
7417	.40	7496	.80	75108A	.35	74L73	.40	74LS10	.25		
7420	.15	74100	1.15	75491	.50	74L74	.45	74LS11	.25		
7426	.25	74107	.25	75492	.50	74L75	.55	74LS20	.20		
7427	.25	74121	.35			74L93	.55	74LS21	.25		
7430	.15	74122	.55			74L123	.85	74LS22	.25		
7432	.20	74123	.35	74H00	.15			74LS32	.25		
7437	.20	74125	.45	74H01	.20	74S00	.35	74LS37	.25		
7438	.20	74126	.35	74H04	.20	74S02	.35	74LS38	.35		
7440	.20	74132	.75	74H05	.20	74S03	.25	74LS40	.30		
7441	1.15	74141	.90	74H08	.35	74S04	.25	74LS42	.65		
7442	.45	74150	.85	74H10	.35	74S05	.35	74LS51	.35		
7443	.45	74151	.65	74H11	.25	74S08	.35	74LS74	.35		
7444	.45	74153	.75	74H15	.45	74S10	.35	74LS86	.35		
7445	.65	74154	.95	74H20	.25	74S11	.35	74LS90	.55		
7446	.70	74156	.70	74H21	.25	74S20	.25	74LS93	.55		
7447	.70	74157	.65	74H22	.40	74S40	.20	74LS107	.40		
7448	.50	74161	.55	74H30	.20	74S50	.20	74LS123	1.00		
7450	.25	74163	.85	74H40	.25	74S51	.25	74LS151	.75		
7451	.25	74164	.60	74H50	.25	74S64	.15	74LS153	.75		
7453	.20	74165	1.10	74H51	.25	74S74	.35	74LS157	.75		
7454	.25	74166	1.25	74H52	.15	74S112	.60	74LS164	1.00		
7460	.40	74175	.80	74H53J	.25	74S114	.65	74LS193	.95		
7470	.45			74H55	.20			74LS367	.75		
7472	.40							74LS368	.65		

LINEARS, REGULATORS, etc.

MCT2	.95	LM320T5	1.65	LM340K15	1.25	LM723	.40
8038	3.95	LM320T12	1.65	LM340K18	1.25	LM725N	2.50
LM201	.75	LM320T15	1.65	LM340K24	1.25	LM739	1.50
LM301	.45	LM324N	1.25	78L05	.75	LM741 (8-14)	.25
LM308 (Mini)	.95	LM339	.75	78L12	.75	LM747	1.10
LM309H	.65	7805 (340T5)	.95	78L15	.75	LM1307	1.25
LM309K (340K-5)	.85	LM340T12	.95	78M05	.75	LM1458	.65
LM310	.85	LM340T15	.95	LM373	2.95	LM3900	.50
LM311D (Mini)	.75	LM340T18	.95	LM380 (8-14 PIN)	.95	LM75451	.65
LM318 (Mini)	1.75	LM340T24	.95	LM709 (8, 14 PIN)	.25	NE555	.35
LM320K5 (7905)	1.65	LM340K12	1.25	LM711	.45	NE556	.85
LM320K12	1.65					NE565	.95
						NE566	1.25
						NE567	.95

9000 SERIES

9301	.85	95H03	1.10
9309	.35	9601	.20
9322	.65	9602	.45

MICRO'S, RAMS, CPU'S, E-PROMS

74S188	3.00	8214	8.95
1702A	4.50	8224	3.25
MM5314	3.00	8228	6.00
MM5316	3.50	8251	8.50
2102-1	1.45	8255	10.50
2102L-1	1.75	8T13	1.50
2114	9.50	8T23	1.50
TR1602B	3.95	8T24	2.00
TMS 4044-	9.95	8T97	1.00
		2107B-4	4.95
8080	8.95	2708	9.50
8212	2.95	Z80 PIO	8.50

INTEGRATED CIRCUITS UNLIMITED

19

7889 Clairemont Mesa Boulevard, San Diego, California 92111

(714) 278-4394 *(Calif. Res.)*

All orders shipped prepaid — No minimum

Open accounts invited — COD orders accepted

Discounts available at OEM Quantities — California Residents add 6% Sales Tax

All IC's Prime/Guaranteed. All orders shipped same day received.

24 Hour Toll Free Phone 1-800-854-2211 — **American Express / BankAmericard / Visa / MasterCharge**

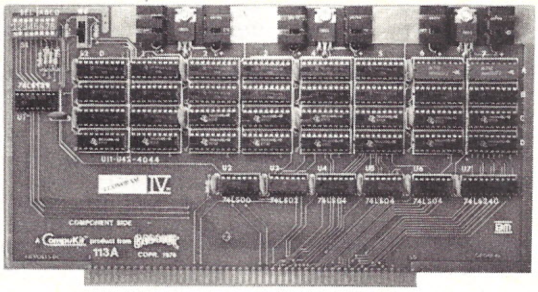
SPECIAL DISCOUNTS

Total Order	Deduct
$35 - $99	10%
$100 - $300	15%
$301 - $1000	20%

quantities of quality!

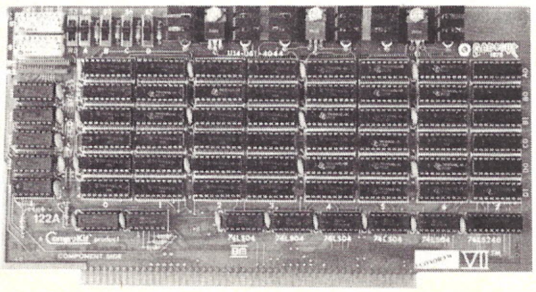
MEET THE ECONORAM FAMILY . . .

These **static** memory kits (one for the H8 buss, all others S-100 compatible) deliver outstanding performance at prices even the dynamics can't match. What others consider "extras" we consider necessities, such as full buffering on all lines, reliable DMA, sockets for all ICs, gold-plated card fingers, prime ICs . . .and all the other signs of quality that make up an Econoram.® **No matter what machine you use, we want to be your memory supplier: and we know the best way to do that is to offer a superior product at the lowest possible price.**



NEW! 16K × 8 ECONORAM IV™ KIT ($329)

Guaranteed current consumption under 2000 mA. Manual write protect switches for 4K blocks; use with or without phantom line. Fully static. Comes with sockets and bypass caps soldered in place for easy assembly. Add $35 for assembled/tested.



NEW! 24K × 8 ECONORAM VII™ KIT ($490)

Our densest board is your best value in 24K memory. Current consumption under 2500 mA; configured as two 4K blocks and two 8K blocks with independent manual write protect switches for each block. Use with or without phantom lines. comes with sockets and bypass caps soldered in place for easy assembly. Add $35 for assembled/tested.

8K × 8 ECONORAM II™ KIT ($135)

A truly cost-effective package that has drawn raves from both owners and reviewers (see the 1/78 Kilobaud for an example). If you have the space in your motherboard, there's no better way to get 24K of memory than taking advantage of our quantity offer (3 kits for $375). Add $20 to single kit price for assembled/tested.

H8 COMPATIBLE ECONORAM VI™ KIT ($235)

12K × 8 for the H8, with the same features that have made our S-100 boards so popular. Additionally, all sockets and bypass capacitors are already soldered in place so you can get right into the best part of kit building.

11 SLOT MOTHERBOARD "UNKIT" ($90)

Those who recognize value love our 10/11 slot Motherboard Kit. Now we've made it even better by pre-soldering all 11 edge connectors in place to take the tedium out of assembly. Includes our much-copied active termination circuitry for optimum buss line characteristics, as well as all edge connectors and plenty of bypass capacitors.

18 SLOT MOTHERBOARD KIT ($124)

Similar to our 11 slot version; includes 18 edge connectors and active termination circuitry. Unlike the 11 slot version, however, the edge connectors are not pre-soldered into place.

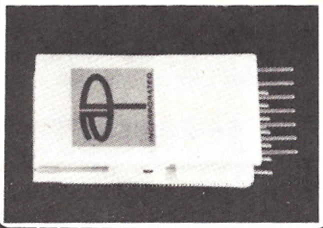
CPU POWER SUPPLY KIT ($50)

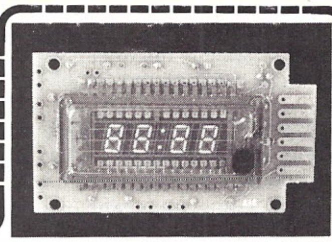
Here is an economical power supply for small computer systems or digital/analog bench work. Delivers +5V @ 4A with crowbar overvoltage protection; also gives ±12V @ ½A per side. Adjustable negative bias supply, 5-10V @ 10 mA. All in all, if you need a small power supply, you can't beat the performance or the price.

TRS-80 16K CONVERSION KIT

This kit contains 8 uPD416 1 × 16K dynamic memories and instructions on converting your 4K TRS-80 to a 16K machine. **You could pay up to $290 elsewhere, but our kit is only $190!**

SOMETHING TO MAKE LIFE EASY: We carry AP test clips for both 14 pin and 16 pin ICs. Gold plated wiping action; sturdy pins for scope probes; also removes ICs from sockets without damage. **Model TC-14 (14 pin): $4.50; Model TC-16 (16 pin): $4.75.** We also carry the A.C.E. **201K breadboarding kit** (with 1,032 solderless plug-in tie point capacity) for only **$24.95.**





MA1003 CLOCK MODULE

Sure, this is a computer magazine; but we just couldn't resist telling you about this beautiful clock. It's a complete module, requiring only time set switches and a source of 12V DC for operation. Built-in time base — ideal for car clocks, or connect to your computer's 12V supply. 4 digit, fluorescent blue readouts. Only **$16.50 each or 3 for $46.**

UNELECTRONIC SPECIAL: LEATHER DICE CUP, $2.50

It doesn't blink, flash, or communicate, but it's a beauty for dice players. Limited quantity. This is *not* a joke!

YES, WE HAVE COMPONENTS . . .not just some components, either, but a truly wide-range selection. From resistors to sophisticated ICs to experimenter and hobbyist items, before you do any shopping check with us . . . you'll be glad you did.

TERMS: Please allow up to 5% for shipping; excess refunded. Add $1 handling for orders under $10. Cal res add tax. COD OK with street address for UPS. For VISA® /Mastercharge® orders call our order desk (24 hrs) at **(415) 562-0636.** Prices good through cover month of magazine.

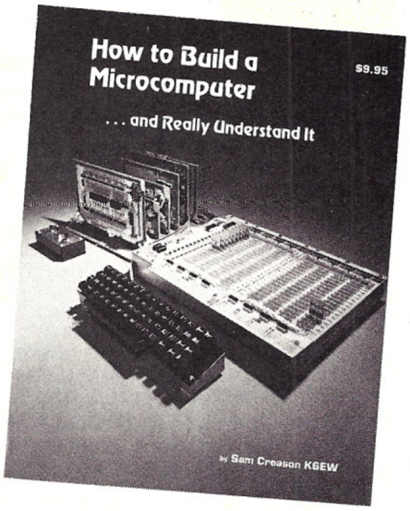
G4

GODBOUT

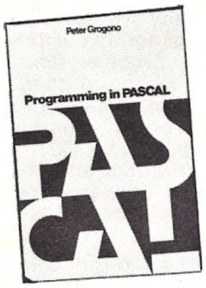
BILL GODBOUT ELECTRONICS
BOX 2355, OAKLAND AIRPORT, CA 94614

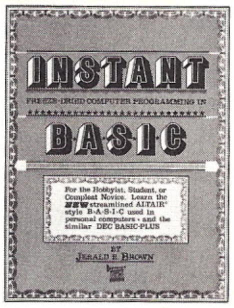
NEW ADDITIONS TO THE KB BOOK NOOK

SPECIAL:
PRE-PUBLICATION OFFER
(orders will be filled on a first-in, first-out basis)



● **HOW TO BUILD A MICROCOMPUTER—AND REALLY UNDERSTAND IT**—BK7325—by Sam Creason. The electronics hobbyist who wants to build his own microcomputer system now has a practical "How-To" guidebook. Sam Creason's book is a combination technical manual and programming guide that takes the hobbyist step-by-step through the design, construction, testing, and debugging of a complete microcomputer system. This book is a must reading for anyone desiring a true understanding of small computer systems. $9.95.*





● **PROGRAMMING IN PASCAL**—BK1140—by Peter Grogono. The computer programming language PASCAL was the first language to embody in a coherent way the concepts of structured programming which had been defined by Edsger Dijkstra and C. A. R. Hoare. As such it is a landmark in the development of programming languages. PASCAL was developed by Niklaus Wirth in Zurich; it is derived from the language ALGOL 60 but is more powerful and easier to use. PASCAL is now widely accepted as a useful language that can be efficiently implemented, and as an excellent teaching tool. This book is intended for people who want to write programs in PASCAL. It does not assume knowledge of any other programming language and it is therefore suitable for an introductory course. $9.95.*

● **MICROPROCESSOR LEXICON—ACRONYMS AND DEFINITIONS**—BK1137—Compiled by the staff of SYBEX is a convenient reference in pocket size format. Sections include acronyms and definitions, part numbers and their definitions, S-100 signals, RS232 signals, IEEE 499 signals, microcomputer and microprocessors, JETDS summary (military) and a code conversion table. $2.95.*

● **INSTANT BASIC**—BK1131—by Jerald R. Brown. For the personal computer enthusiast or the user of DEC's BASIC PLUS language, here, finally, is a new book to teach you BASIC. It teaches BASIC to beginners using interesting programming ideas and applications that will be easily understood by the home computer programmer. BASIC PLUS users know that the two languages are very similar, so this book can be used by them as well. This is an "active participation" workbook, designed to be used with your home computer so you can learn by doing! Ideas are slowly introduced in a nonmathematical context so the beginner can quickly learn good programming techniques. $6.00.*

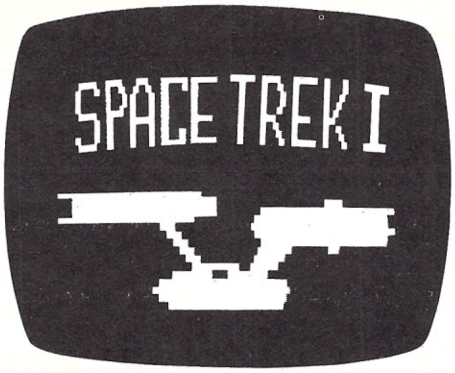
● **YOUR HOME COMPUTER**—BK1172—by James White, is an introduction to the world of personal microcomputing. Written in clear and understandable language, this book tells you everything you want to know about home computing and gives the computer novice a painless introduction to microcomputer technology and terminology, beginning with what computers are and how they work. This basic book requires no prior knowledge or experience in electronics or computing.

It provides information about home computer kits, guidelines for selecting and building your own microcomputer, how to use your home computer and what you can do with it, lists of computer stores, clubs, periodicals, and answers to many more of your questions about microcomputers and the jargon surrounding the personal computing scene today. $6.00.

INSTANT SOFTWARE FROM KILOBAUD

● **INSTANT SOFTWARE FROM KB** The following are the first of many prerecorded programs that will be available from **Kilobaud.** Each cassette is accompanied by a booklet that includes a hard-copy version of the program as well as a description of intended use. All cassettes are available at $7.95* unless otherwise listed.

● **BASIC LUNAR LANDER/INTERMEDIATE LUNAR LANDER**—SP0101—by Ed Juge (for TRS-80™) Your automatic landing system has malfunctioned and you are forced to make a manual landing. Taking control at 15,000 feet, the pilot's success requires precise manipulation of the lunar lander's retro-rockets. After mastering the basic game, the able commander will be taxed by the difficult intermediate program. $7.95.*



● **SPACE TREK**—SP0102—by Ed Juge (for the TRS-80) This Space Trek game will challenge the most skillful and careful player. As commander of the Enterprise, you must defend the galaxy against enemy Klingon warships using your arsenal of phasers and photon torpedoes to best advantage. The Enterprise is also equipped with impulse power for moving within the quadrant, and warp power for jumping from quadrant to quadrant. $7.95.*

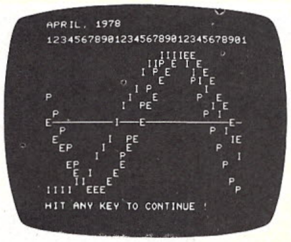
● **SPACE TREK**—SP0103—by Raymond Schaefer (for the TRS-80) Another version of this popular game, Schaefer's program requires the player to defend his galaxy against a Klingon expeditionary force. The galaxy is divided into 4 x 4 quadrants, each further subdivided into 16 sectors. An on-board computer warns of appraoaching danger and of ominously low shield power. $7.95.*

continued . . .

* Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to: Kilobaud Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information. For orders under $10.00, add $1.00 shipping and handling.

KB book nook

Instant Software can't . . .



●**PERSONAL WEIGHT CONTROL/ BIORHYTHMS**—SP0105—by Microcosm (for the PET™) Here's a program that will let you chart any individual's biorhythm cycle, showing physical, emotional, and intellectual highs and lows. If your neighbors find out you've got it, they will never leave you in peace. **PERSONAL WEIGHT CONTROL** is a must for the diet-conscious. Once your ideal weight has been determined, the program will calculate the correct daily caloric intake necessary to reach your goal over a certain time period. $7.95.*

●**BEGINNER BACKGAMMON/KENO**—SP0104 —by Fred Schilds and Ed Juge (for the TRS-80™) You will never be in need of a backgammon companion again! Schilds's program is designed for the newcomer to the game, but the computer's steady play will challenge the careless expert. **KENO** is a popular Las Vegas gambling game. As in many games of chance, the odds are against you, but when you win you can win big. $7.95.*

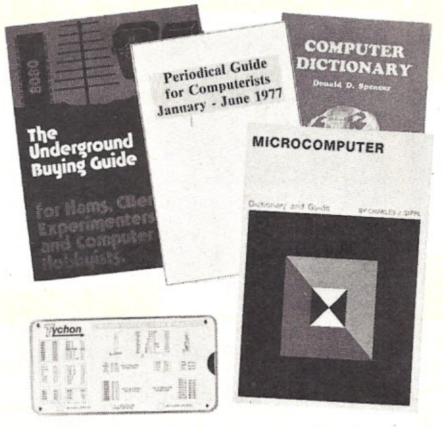
●**MORTGAGE WITH PREPAYMENT OPTION/FINANCIER**—SP0106—by Microcosm (for the PET™) Before you go to the bank, calculate your mortgage payment schedule and find out what you can afford. You will learn how much prepayments save, and can easily compare the true costs of variations in interest. **FINANCIER** is actually three programs in one and is guaranteed to make you a financial wizard. In part I, you'll learn to tell which investments will pay off. Part II calculates annual depreciation schedules, providing rates, amounts, and salvage values, which is great when tax time rolls around. The final section figures costs of borrowing, payment terms, and remaining balances on loans of all sizes. $7.95.*

general

●**COMPUTER DICTIONARY**—BK1013—by Donald D. Spencer. Defines words and acronyms used by computerists in a clear, easy to understand style. This reference is a must for the individual getting started in the world of microcomputers. $5.95.*

●**MICROCOMPUTER DICTIONARY**—BK1034—This microcomputer dictionary fills the need to become quickly acquainted with the terminology and nomenclature of the revolution in computers. There is also a comprehensive electronics/computer abbreviations and acronyms section. $15.95.*

●**THE UNDERGROUND BUYING GUIDE**—BK1067—Here is a handy guide for the electronics enthusiast. Over 600 sources of equipment and literature are provided. Cross-referenced for ease of use. Electronic publishing houses are also listed. $5.95.*



●**1976 PERIODICAL GUIDE FOR COMPUTERISTS**—BK1041—is a 20-page book which indexes over 1,000 personal computing articles for the entire year of 1976 from *Byte, Creative Computing, Digital Design, Dr. Dobbs Journal, EDN, Electronic Design, Electronics, Interface Age, Microtrek, Peoples' Computer Company, Popular Electronics, QST, Radio Electronics, SCCS Interface* and *73 Amateur Radio.* $3.00.* New January-June 1977 Edition BK1141—(includes Kilobaud)—$3.00.*

●**TYCHON'S 8080 OCTAL CODE CARD**—CC1066—Slide rule-like aid for programming and debugging 8080 software contains all the mnemonics and corresponding octal codes. Also available, Tychon's *8080 Hex Code Card,* same as above only has hex codes instead of octal—CC1065—$3.00 each.*

amateur radio books

●**NOVICE STUDY GUIDE**—SG7357—The most complete Novice study guide available. It is brand new. This is not only invaluable for anyone wanting to get started in amateur radio, but also it is about the only really simple book on the fundamentals of electricity and electronics. $4.95.*

●**GENERAL CLASS STUDY GUIDE**—SG7358—Takes over on theory where the Novice book leaves off. You'll need to know the electronic theory in this to work with computers and you'll not find an easier place to get the information. $5.95.*

●**SSTV HANDBOOK** This excellent book tells all about it, from its history and basics to the present state-of-the-art techniques. Hardbound—BK7354—$7.* Softbound—BK7355—$5.*

●**VHF ANTENNA HANDBOOK**—BK7368—This new handbook details the theory, design and construction of hundreds of different VHF and UHF antennas. Packed with fabulous antenna projects you can build. $4.95.

●**WEATHER SATELLITE HANDBOOK**—BK7370—Simple equipment and methods for getting good pictures from the weather satellite. Dr. Taggart WB8DQT. $4.95.*

●**THE NEW RTTY HANDBOOK**—BK7347—is a brand new 1977 edition and the only up-to-date RTTY book available. The state-of-the-art has been changing radically and has made all previous RTTY books obsolete. It has the latest circuits, great for the newcomer and the expert alike. $5.95.*

computer games

●**SCELBI'S FIRST BOOK OF COMPUTER GAMES**—BK1048—Need a game for your 8008 or 8080 microprocessor? Has three popular games, "Space Capture," "Hexpawn," and "Hangman." Complete flowcharts, logic description, program listing, and instructions are provided. A must for the game freak! $14.95.*

●**BASIC COMPUTER GAMES**—BK1074—Okay, so once you get your computer you and running in BASIC, then what? Then you need some programs in BASIC, that's what. This book has 101 games for you, from very simple to real buggers. You get the games, a description of the games, the listing to put in your computer and a sample run to show you how they work. Fun. Any one game will be worth more than the price of the book for the fun you and your family will have with it. $7.50.*

●**WHAT TO DO AFTER YOU HIT RETURN**—BK1071—PCC's first book of computer games . . . 48 different computer games you can play in BASIC . . . programs, descriptions, many illustrations. Lunar Landing, Hammurabi, King, Civel 2, Qubic 5, Taxman, Star Trek, Crash, Market, etc. $8.00.*

* Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to: Kilobaud Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information. For orders under $10.00, add $1.00 shipping and handling.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

KB book nook

beginner's·introductory

● **THE STORY OF COMPUTERS**—BK1056 —by Donald D. Spencer, is to computer books what *Dick and Jane* is to novels . . . elementary, gives the non-computerist a fair idea of what the hobbyist is talking about when he speaks computer lingo. Attempts to explain what computers are and can do to a spouse, child or any un-electronics-minded friend. $4.95.*

● **HOBBY COMPUTERS ARE HERE!**—BK7322—If you want to come up to speed on how computers work . . . hardware and software . . . this is an excellent book. It starts with the fudamentals and explains the circuits, the basics of programming, along with a couple of TVT construction projects, ASCII-Baudot, etc. This book has the highest recommendations as a teaching aid for newcomers. $4.95.*

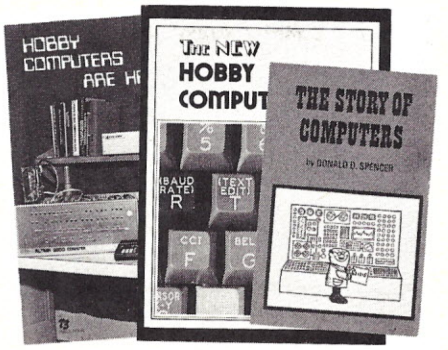
● **THE NEW HOBBY COMPUTERS**—BK7340—This book takes it from where "**Hobby Computers Are Here!**" leaves off, with chapters on Large Scale Integration, how to choose a microprocessor chip, an introduction to programming, low cost I/O for a computer, computer arithmetic, checking memory boards, a Baudot monitor/editor system, an audible logic probe for finding those tough problems, a ham's computer, a computer QSO machine . . . and much, much more! $4.95*

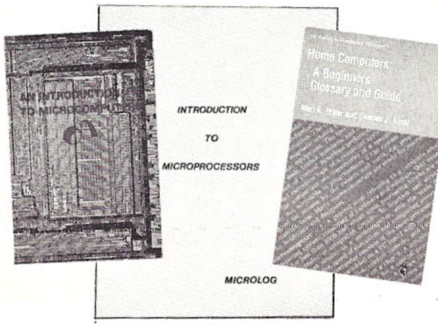
● **HOME COMPUTERS: 2^{10} QUESTIONS & ANSWERS** by Rich Didday. Two books aimed exclusively at the novice computer hobbyist/home computer user. Written in a rather unusual style which has a beginner asking questions which are answered by a person with a substantial background in computers and personal computing. The questions are just the kind beginners come up with . . . and the answers are presented in easy-to-understand terms (usually with a diagram to illustrate the point). Both the hardware and software aspects of home computing are covered from A to Z. An index in both books makes them ideal as reference material *for anyone*. Volume I: Hardware—BK1023 —$7.95*; Volume II: Software—BK1024—$6.95.*

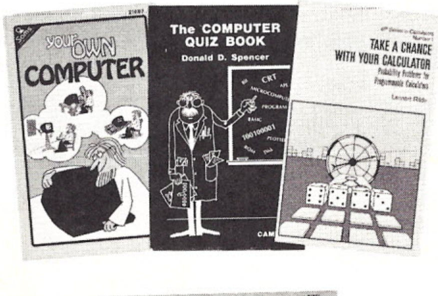
● **MICROCOMPUTER PRIMER**—BK1035 —by M. Waite and M. Pardee. Describes basic computer theory, explains numbering systems, and introduces the reader to the world of programming. Describes the world of microcomputing in "real world" terminology. $7.95.*

● **A STEP BY STEP INTRODUCTION TO 8080 MICROPROCESSOR SYSTEMS**—BK1054—by D. L. Cohn & J. L. Melsa. The step-by-step presentation does not require any computer or electronics background, so anyone who is interested can follow it. However, the book is not only intended for beginners. Engineers and technicians who are familiar with electronics will find the software descriptions valuable in updating their skills. Computer professionals will find the detailed treatment of the 8080 architecture and intruction set useful. $7.95.*

● **HOW TO BUY & USE MINICOMPUTERS AND MICROCOMPUTERS**—BK1025—by Wm. Barden, Jr. This book discusses these smaller brethren of computers and show how the reader can become a part of the revolution—how he can own and use a functioning computer system in his home to do a variety of practical or recreational tasks. $9.95.*







● **MICROPROCESSORS FROM CHIPS TO SYSTEMS**—BK1036—by Rodnay Zaks, is a complete and detailed introduction to microprocessors and microcomputer systems. No preliminary knowledge of computers or microprocessors is required to read this book, although a basic engineering knowledge is naturally an advantage. Intended for all wishing to understand the concepts, techniques and components of microprocessors in a short time. $9.95.*

● **INTRODUCTION TO MICROPROCESSORS**—BK1032—by Charles Rockwell of MICROLOG is an ideal reference for the individual desiring to understand the hardware aspects of microprocessor systems. Describes the hardware details of computer devices in terms the beginner can understand, instead of treating the micro chip as a "black box." Specific systems are not described and programming is only briefly discussed. $17.50 US and Canada, $20 elsewhere.*

● **AN INTRODUCTION TO MICROCOMPUTERS, VOLS. 1 AND 2** by Adam Osborne Associates, are references dealing with microcomputer architecture in general and specifically with details about most of the common chips. These books are not software-oriented, but are invaluable for the hobbyist who is into building his own interfaces and processors. Volume I is dedicated to general hardware theory related to micros, and Volume II discusses the practical details of each micro chip. (Detailed review in Kilobaud #2) Volume I—BK1030—$7.50*; Volume II—BK1031—$12.50.*

● **HOME COMPUTERS: A BEGINNER'S GLOSSARY AND GUIDE**—BK1022—this book is intended as a quick reference source for beginners. Included is a general introduction to microcomputers, a simple application & sample system, the history of microcomputers & their uses, and an introduction to some actual equipment. A chapter on number systems includes a number conversion chart, binary arithmetic from conversions to divisions, and a discussion of octal and hexadecimal numbers. A good background to read technical literature and computer equipment specifications. $6.95.*

● **YOUR OWN COMPUTER**—BK1072—by M Waite and M. Pardee. The personal computer has been touted as the next consumer product. But most individuals still wonder why. Much technical material has been written but there is little material for the average individual without an extensive background in electronics. This book removes the stigma of complexity that surrounds the computer and has succeeded in providing a simple, easy-to-understand guide to these units. $1.95.*

● **TAKE A CHANCE WITH YOUR CALCULATOR**—BK1002—Lennart Rade wrote this book to help you to discover the world of probability with your programmable calculator. You will need NO previous experience either in probability theory or in programming to learn both from this book. $8.95.*

● **THE COMPUTER QUIZ BOOK**—BK1112 —by Donald D. Spencer, is written for readers who would like to test themselves on basic computer concepts. It may be used effectively by anyone interested in checking their knowledge of computer concepts, $5.95.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to:
Kilobaud Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information.
For orders under $10.00, add $1.00 shipping and handling.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

KB book nook KB book nook

software • programming

● **DISCOVERING BASIC—A PROBLEM SOLVING APPROACH**—BK1017—by Robert E. Smith, deals with progressively more complex problems which allow the reader to discover the vocabulary of BASIC language as he develops skill and confidence in putting it to work. Clear and concise explanations. Problems used cover a wide range of interests—insurance, geometry, puzzles, economics, etc. $6.85.

● **BASIC NEW 2ND EDITION**—BK1081—by Bob Albrecht. Self-teaching guide to the computer language you will need to know for use with your microcomputer. This is one of the easiest ways to learn computer programming. $4.95.*

● **A QUICK LOOK AT BASIC**—BK1043—by Donald D. Spencer. A perfect reference for the beginning programmer. Assumes that the reader has no previous programming experience and is a self-teaching guide for the individual desiring to learn the fundamentals of BASIC. $4.95.*

● **MY COMPUTER LIKES ME . . . WHEN I SPEAK BASIC**—BK1039—An introduction to BASIC . . . simple enough for kids. If you want to teach BASIC to anyone quickly, this is the way to go. $2.00.*

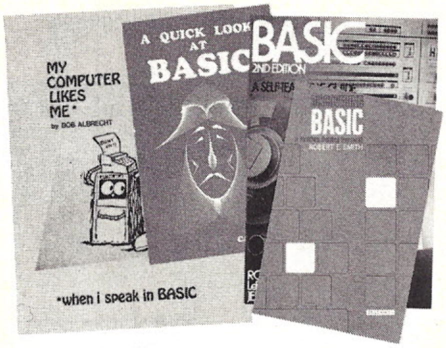
● **FUN WITH COMPUTERS AND BASIC**—BK1021—by Donald D. Spencer, contains an easy-to-understand explanation of the BASIC Programming Language and is intended for persons who have had no previous exposure to computer programming. Over half the book is devoted to problems using games, puzzles, and mathematical recreations. A superior book for self-teaching and learning computer programming. $6.95.*

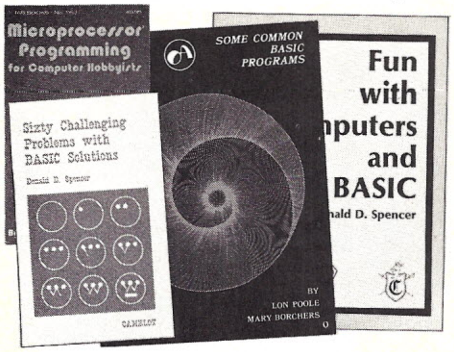
● **SIXTY CHALLENGING PROBLEMS WITH BASIC SOLUTIONS**—BK1073—by Donald Spencer, provides the serious student of BASIC programming with interesting problems and solutions. No knowledge of math above algebra required. Includes a number of game programs, as well as programs for financial interest, conversions and numeric manipulations. $6.95.*

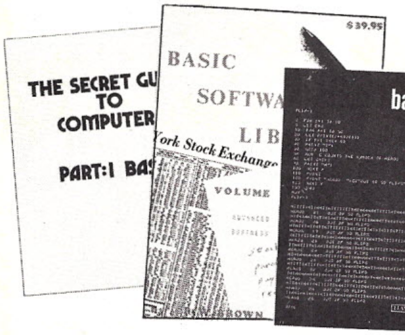
● **THE SECRET GUIDE TO COMPUTERS** Parts 1, 2, and 3 by Russ Walter. **Part One** describes computers in general, and after reading for ten minutes you will be writing simple BASIC programs! **Part Two** discusses computer applications. It's one thing to master the syntax of the language such as BASIC and another to solve problems using the new tool. **Part Three** describes programming languages. Ever heard of APL and QLISP? BASIC is not the only language used to program computers. 7th Edition. Part I—BK1050—$2.75*; Part II—BK1051—$2.50*; Part III—BK1052—$3.50.*

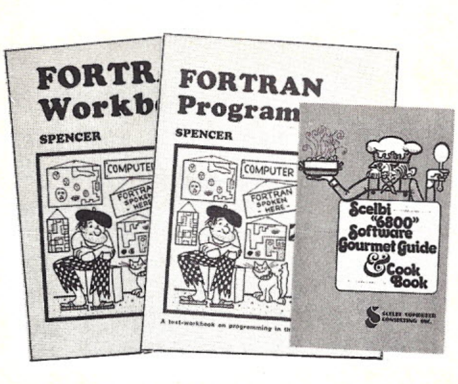
● **SOME COMMON BASIC PROGRAMS**—BK1053—published by Adam Osborne & Associates, Inc. Perfect for non-technical computerists requiring ready-to-use programs. Business programs, plus miscellaneous programs. Invaluable for the user who is not an experienced programmer. All will operate in the stand-alone mode. $8.50 paperback.*

● **ADVANCED BASIC**—BK1000—Applications and problems by James Coan is for those who want to extend their expertise with BASIC. Offers advanced techniques and applications. $7.95.*









● **SCIENTIFIC RESEARCH INSTRUMENTS' BASIC SOFTWARE LIBRARY** is a complete do-it-yourself kit. Written in everybody's BASIC immediately executable in ANY computer with 4K, no other peripherals needed. **Volume I** contains business and recreational programs and is 300 pages. **Volume II** is 260 pages and contains math, engineering, statistics and plotting programs. **Volume III** contains money managing, advanced business programs such as billing, A/R, inventory, payroll, etc. **Volume IV** contains general purpose programs like loans, rates, retirement, plus games: Poker, Enterprise (take charge while Capt. Kirk is away), Football and more! **Volume V** is filled with experimenter's programs including games, pictures and misc. problems like "logic." Volume I—LB1002—& Volume II—LB1003—$24.95* each, Volume III—LB1004—$39.95*, Volume IV—LB1005—& Volume V—LB1006—$9.95* each, Volume VI—LB1007—$49.95*, Volume VII—LB1008—$39.95.*

● **MICROPROCESSOR PROGRAMMING FOR COMPUTER HOBBYISTS**—BK1038—by Neill Graham is for the hobbyist interested in intermediate and advanced techniques of programming and data structuring. Written to take up where the computer manufacturers' instruction manuals and the introductory programming language texts leave off. $8.95.*

● **8080 PROGRAMMING FOR LOGIC DESIGN**—BK1078—Ideal reference for an in-depth understanding of the 8080 processor. Application-oriented and the 8080 is discussed in light of replacing conventional, hard-wired logic. Practical design considerations are provided for the implementation of an 8080-base control system. $8.50.*

● **8080 SOFTWARE GOURMET GUIDE AND COOKBOOK**—BK1102—If you have been spending too much time developing simple routines for your 8080, try this new book by Scelbi Computing and Robert Findley. Describes sorting, searching, and many other routines for the 8080 user. $9.95.*

● **6800 PROGRAMMING FOR LOGIC DESIGN**—BK1077—Oriented toward the industrial user, this book describes the process by which conventional logic can be replaced by a 6800 microprocessor. Provides practical information that allows an experimenter to design a complete micro control system from the "ground up." $8.50.*

● **6800 SOFTWARE GOURMET GUIDE & COOKBOOK**—BK1075—If you have been spending too much time developing routines for your 6800 microprocessor, try the new book by Scelbi Computing and Robert Findley. Describes sorting, searching, and many other routines for the 6800 user. $10.95.*

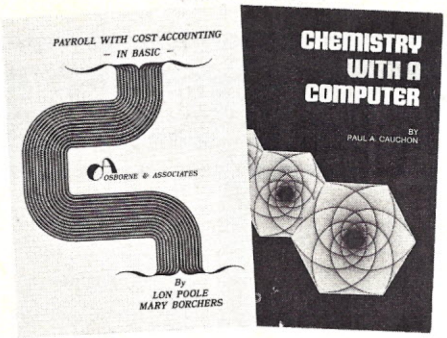
● **FORTRAN PROGRAMMING**—BK1019—by Donald Spencer. FORTRAN was designed for complex numeric calculations, and possesses extended I/O capability. It is easily learned, as it is an English-like computer language. $7.95.*

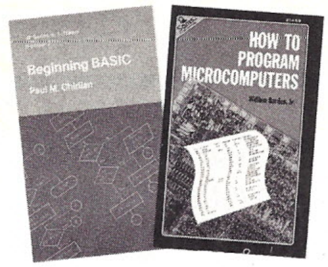
● **FORTRAN WORKBOOK**—BK1020—by Donald Spencer. Provides practical examples and problems to solve. Flowcharting is also discussed. Almost all micros support BASIC—it won't be long before FORTRAN is commonplace. $4.95.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to:
Kilobaud Book Department ● Peterborough NH 03458. Be sure to include check or detailed credit card information.
For orders under $10.00, add $1.00 shipping and handling.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

software•programming
. . . continued





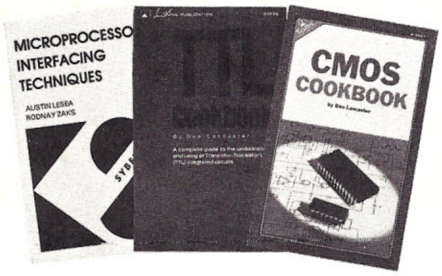
•PAYROLL WITH COST ACCOUNTING—IN BASIC—BK1001—by L. Poole & M. Borchers, includes program listings with remarks, descriptions, discussions of the principle behind each program, file layouts, and a complete user's manual with step-by-step instructions, flow charts, and simple reports and CRT displays. Payroll and cost accounting features include separate payrolls for up to 10 companies, time-tested interactive data entry, easy correction of data entry errors, job costing (labor distribution), check printing with full deduction and pay detail, and 16 different printed reports, including W-2 and 941. $12.50.*

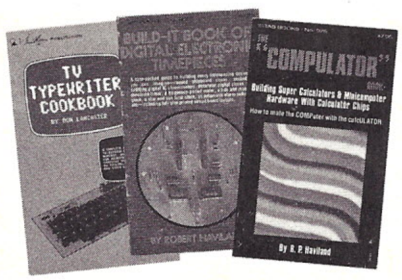
•CHEMISTRY WITH A COMPUTER—BK1010—by Paul A. Cauchon, contains a collection of tutorial, simulation and problem-generation computer programs. Usable with almost any chemistry course in the high school or college level. $9.95.*

•HOW TO PROGRAM MICROCOMPUTERS—BK1027—by Wm. Barden, Jr. Here is a guide to assembly language programming of the Intel 8080, Motorola MC6800, and MOS Technology MCS6502 microprocessors. It is written especially for beginning programmers with hobbyist microcomputers based on one of these three chips. The topics covered range from data manipulations at the bit level up to data handling of tables and lists, and from simple adds and subtracts up to floating-point operations. $8.95.*

•BEGINNING BASIC—BK1007—by Dr. Paul Chirlian, provides a really basic BASIC that covers all the topics in simple, easy-to-understand language. Nothing is left out, everything is presented in clear, step-by-step fashion. This book will make a good BASIC programmer of any reader. $9.95.*

hardware





•MICROPROCESSOR INTERFACING TECHNIQUES—BK1037—by Austin Lesea & Rodnay Zaks will teach you how to interconnect a complete system and interface it to all the usual peripherals. It covers hardware and software skills and techniques, including the use and design of model buses such as the IEEE 488 or S100. $9.95.*

•TTL COOKBOOK—BK1063—by Donald Lancaster. Explains what TTL is, how it works, and how to use it. Discusses practical applications, such as a digital counter and display system, events counter, electronic stopwatch, digital voltmeter, and a digital tachometer. $8.95.

•CMOS COOKBOOK—BK1011—by Don Lancaster. Details the application of CMOS, the low power logic family suitable for most applications presently dominated by TTL. Required reading for every serious digital experimenter! $9.95.*

•TVT COOKBOOK—BK1064—by Don Lancaster, describes the use of a standard television receiver as a microprocessor CRT terminal. Explains and describes character generation, cursor control and interface information in typical, easy-to-understand Lancaster style. $9.95.

•BUILD-IT BOOK OF DIGIAL ELECTRONIC TIMEPIECES—BK1008—by Robert Haviland is a data-packed guide to building every timekeeping device you can imagine: rugged shipboard clocks, second-splitting digital IC chronometers, decorator digital clocks, a precision timer, a frequency-period meter, a tide and moon clock, an automatic alarm setter, etc. Including full-size printed circuit board layouts. $6.95.*

•THE "COMPULATOR" BOOK—BUILDING SUPER CALCULATORS & MINICOMPUTER HARDWARE WITH CALCULATOR CHIPS—BK1012—by R. P. Haviland, provides ideas, design info and printed circuit boards for calculator chip projects, measure time, tie in with a Teletype to create a virtually infinite memory system, and countless other functions. $7.95.*

73 TEST EQUIPMENT LIBRARY

•VOLUME I COMPONENT TESTERS—LB7359—Build your own test equipment and save a bundle (and have a lot of fun). Volume I of the 73 Test Equipment Library shows you how to build and use transistor testers (8 of 'em), three diodes testers, 3 IC testers, 9 voltmeters and VTVMs, 8 ohmmeters, 3 inductance meters, and a raft of other gadgets for checking temperature, crystals, Q, etc. $4.95.*

•VOLUME II AUDIO FREQUENCY TESTERS—LB7360—If you're into audio such as digital cassette recording, RTTY, Baudot vs. ASCII, SSTV, SSB, Touchtone or even hi-fi you'll want to have this book full of home-built test equipment projects. $4.95.*

•VOLUME III RADIO FREQUENCY TESTERS—LB7361—This is of more interest to hams and CBers: test equipment you can build for checking out transmitters and receivers, signal generators, noise generators, crystal calibrators, GDOs, dummy loads. $4.95.*

•VOLUME IV IC TEST EQUIPMENT—LB7362—Become a trouble-shooting wizard. All you need to know about pulse, audio and sync generators, frequency counters, digital component testers, logic probes and more! Plus a cumulative index for all four volumes of the 73 Test Equipment Library. $4.95.

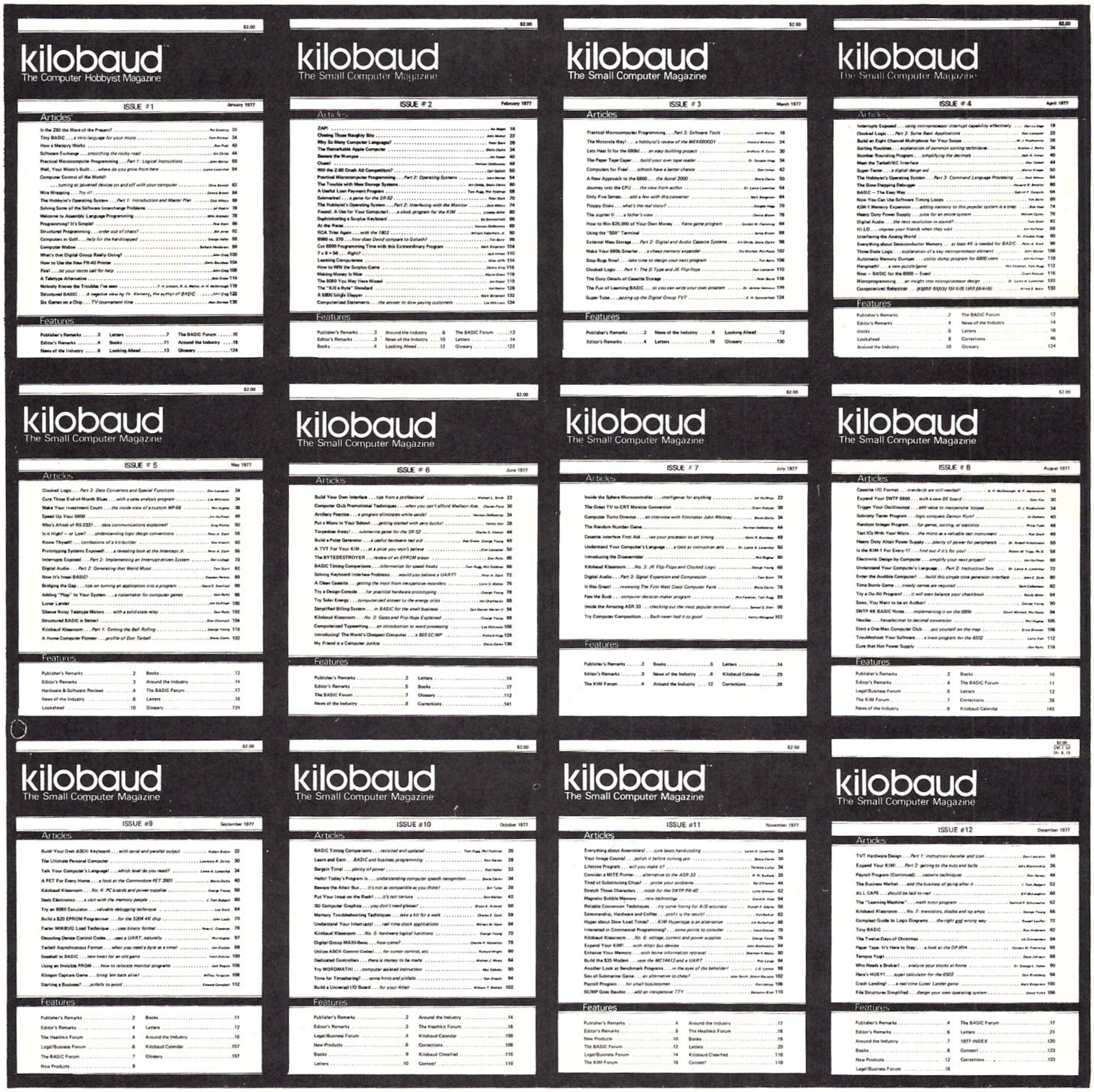
•LSI-11 REFERENCE CARD—CC 2011—A must for Heath H11 users. Contains listings of all instructions by mnemonic and opcode, console ODT commands, trap and interrupt vector locations and more. From Digital Equipment Corporation. $1.00.*

*Use the order card in the back of this magazine or itemize your order on a separate piece of paper and mail to: Kilobaud Book Department • Peterborough NH 03458. Be sure to include check or detailed credit card information. For orders under $10.00, add $1.00 shipping and handling.

FOR TOLL FREE ORDERING CALL 1-800-258-5473

KB book nook KB book nook

1977 KB BACK ISSUES



All issues subject to availability!
Issue #1—$5.00; #2 thru #12—$3 ea.; #13 & #14—$10 ea.
Please send Kilobaud back issues:

___ issues JAN 77-$5
___ issues FEB 77—$3
___ issues MAR 77—$3
___ issues APR 77—$3
___ issues MAY 77—$3
___ issues JUNE 77—$3
___ issues JUL 77—$3
___ issues AUG 77—$3
___ issues SEPT 77—$3
___ issues OCT 77—$3
___ issues NOV 77—$3
___ issues DEC 77—$3
___ issues JAN 78—$10
___ issues FEB 78—$10

TOTAL: ________

Have credit card # handy and call
TOLL FREE (800) 258-5473
Kilobaud • Peterborough NH 03458

KB/8/78

Add $1.00 shipping & handling for all orders under $10.00.

☐ BankAmericard/Visa ☐ Master Charge ☐ American Express

Card # ________ Expiration date ________

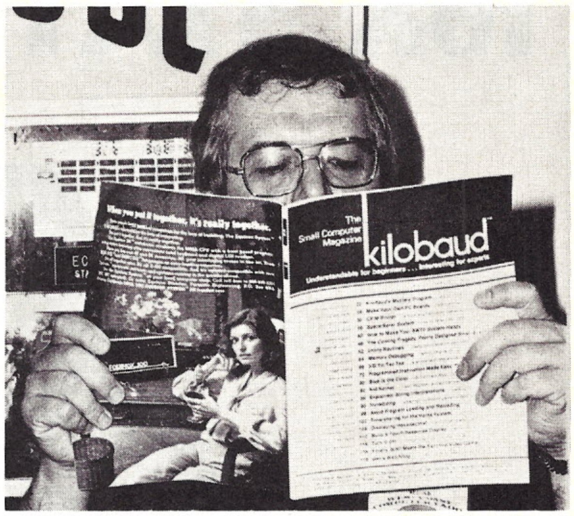
Signature ________

Name ________

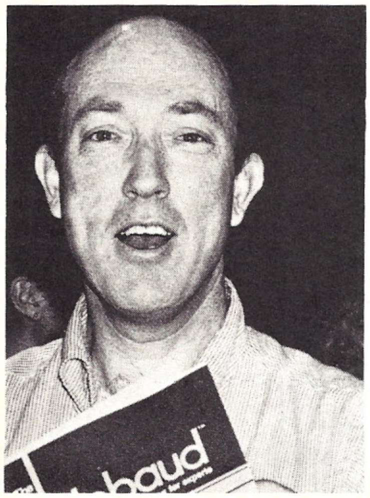
Address ________

City ________ State ____ Zip ____

WHO'S BEHIND THE kilobaud ?



If you don't recognize those beady eyelids looking over a competitor's ad in Kilobaud, then you don't get to many computer shows. Then again, neither does Mr. Z, who prefers to smugly turn out some of the post popular computer boards in the business.



LAST MONTH'S MYSTERY READER

This man behind the Kilobaud was no mystery, surely. It's George Morrow . . . you are either using one of his front panels or know someone who is. He runs Thinker Toys . . . designs for Equinox, Parasitic Engineering and Godbout . . . and probably others. His Wunderbar motherboard is one of the best out. He's one of the first people to treat the S-100 bus like the radio circuit it is . . . and thus one of the first to tame it. If you get a chance to talk with George at any length it will be an experience you'll remember . . . ideas come pouring out almost faster than he can talk.

Subscribe Today!
For Instant Subscriptions Call Our Toll Free Number
1 (800) 258-5473
Have your credit card handy!
Kilobaud • Peterborough NH 03458

MOVING?

Let us know 8 weeks in advance so that you won't miss a single issue of *Kilobaud*.
Attach old label where indicated and print new address in space provided. Also include your mailing label whenever you write concerning your subscription. It helps us serve you promptly.

- ☐ Address change only
- ☐ Extend subscription
- ☐ Enter new subscription
- ☐ 1 year $15.00
- ☐ Payment enclosed (1 extra BONUS issue)
- ☐ Bill me later

If you have no label handy, print OLD address here.

AFFIX LABEL

Name ______________ Call ______
Address ______________
City ________ State ______ Zip ______

print NEW address here:

Name ______________ Call ______
Address ______________
City ________ State ______ Zip ______

• KILOBAUD • PETERBOROUGH NH 03458 •

Radio Shack's personal computer system? This ad just might make you a believer.

You can't beat the 4K system at
$599

TRS-80 "Breakthru"
- TRS-80 microcomputer
- 12" video display
- Professional keyboard
- Power supply
- Cassette tape recorder
- 4K RAM, Level-I BASIC
- 232-page manual
- 2 game cassettes

...or the step-up 16K system at
$899

TRS-80 "Sweet 16"
- Above, except includes 16K RAM

...or the fast 4K/printer system at
$1198

TRS-80 "Educator"
- Above, except includes 4K RAM and screen printer

...or the Level-II 16K/printer/disk system at
$2385

TRS-80 "Professional"
- Above, except includes 16K RAM, disk drive, expansion interface, and Level-II BASIC

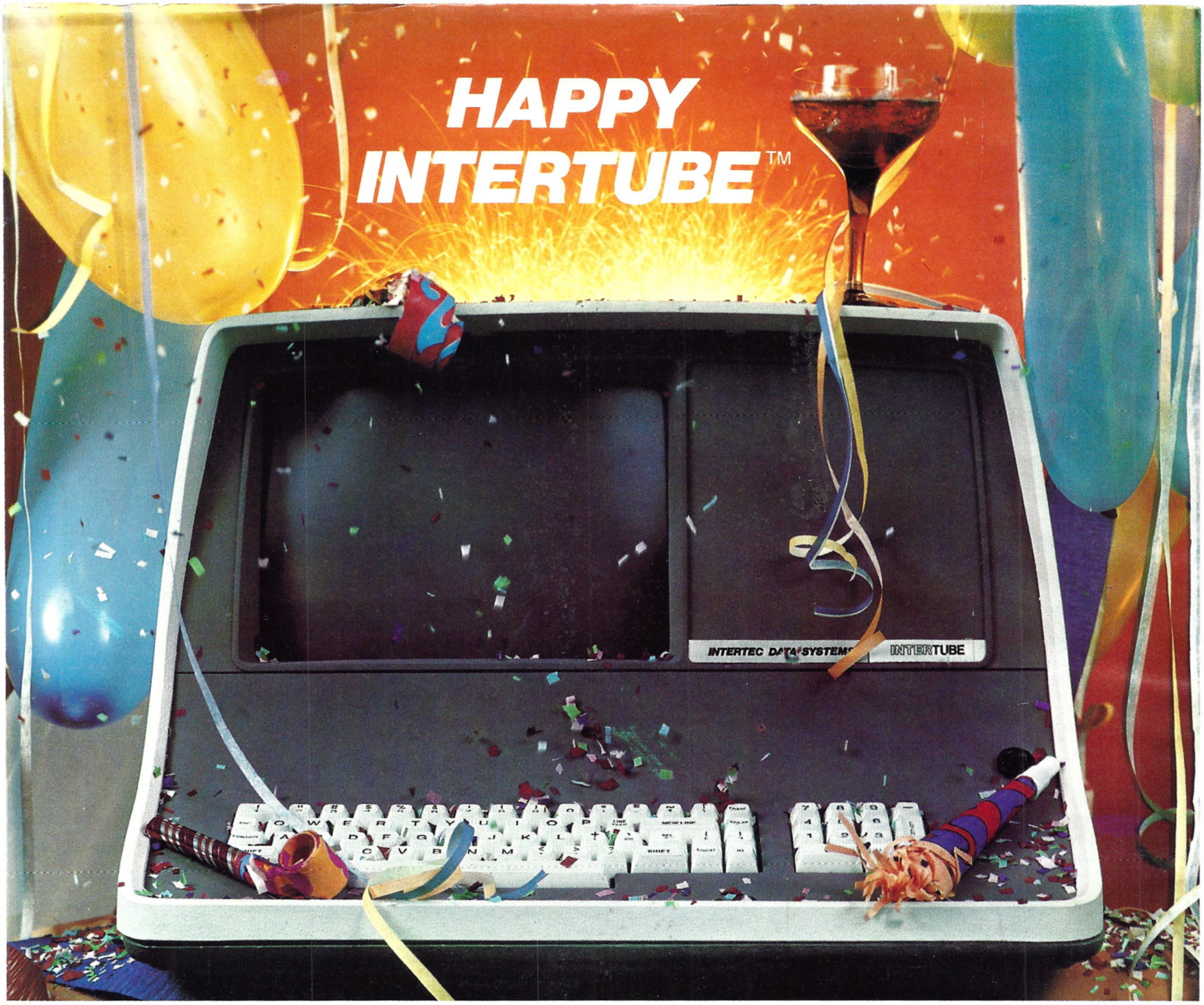
So how are you gonna beat the system that does this much for this little? No way!

...The amazing new 32K/Level-II/2-disk/ line printer system at
$3874

TRS-80 "Business"
- Above, except includes 32K RAM, line printer, and two disk drives

Get details and order now at Radio Shack stores and dealers in the USA, Canada, UK, Australia, Belgium, Holland, France, Japan. Write Radio Shack, Division of Tandy Corporation, Dept. C-052, 1400 One Tandy Center, Fort Worth, Texas 76102. Ask for Catalog TRS-80.

Radio Shack®
The biggest name in little computers



784 Reasons to Celebrate

The party's over for all dumb terminals and a lot of smart ones too. But, at $784 (quantity one), the party's just beginning for Intertec's INTERTUBE.

Standard features to celebrate include a full 24 line by 80 character display, 128 upper and lower case ASCII characters, reverse video, complete cursor addressing and control, an 18 key numeric key pad, special function keys, blinking, a self-test mode, protected fields, line and/or character editing capabilities, eleven special graphic symbols and an RS-232 printer port.

You'll discover even more reasons to celebrate when you sit down in front of an INTERTUBE. Our special AccuDot focusing technique produces crisp, sharp characters on a non-glare screen. INTERTUBE's Z-80 processor assures extreme flexibility with operator oriented features to boost the efficiency of both software and programmers. And, if service is ever required, INTERTUBE's well-designed modular component layout insures quick service by a nationwide factory-trained network providing more than 250 local dealer and service center outlets.

INTERTUBE's combination of price and performance can satisfy your requirements whether it be a sophisticated data entry application or a simple inquiry/response environment. So, there's really no reason to think "dumb" when you can afford to be so smart!

There's an INTERTUBE celebration going on near you. Contact us at one of the numbers below for the name and location of the nearest party. We'll be surprised if you don't come (BYOB). Happy INTERTUBE!

INTERTEC DATA SYSTEMS

Corporate Headquarters
1851 Interstate 85 South
Charlotte, North Carolina 28208
704/377-0300

Eastern Regional Marketing
19530 Club House Road
Gaithersburg, Maryland 20760
301/948-2400

Western Regional Marketing
17952 Sky Park Blvd.
Irvine, California 92714
714/957-0300